NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

TEST AND EVALUATION OF AN IMAGE-MATCHING NAVIGATION SYSTEM FOR A UAS OPERATING IN A GPS-DENIED ENVIRONMENT

by

Keng Siew Aloysius Han

September 2017

Thesis Advisor Oleg Yakimenko
Co-Advisor Ryan Decker

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704–0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202–4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave Blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 2017	Master's Thesis 09-22-2016 to 09-22-2017

4. TITLE AND SUBTITLE

TEST AND EVALUATION OF AN IMAGE-MATCHING NAVIGATION SYSTEM FOR A UAS OPERATING IN A GPS-DENIED ENVIRONMENT

5. FUNDING NUMBERS

6. AUTHOR(S)

Keng Siew Aloysius Han

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Postgraduate School
Monterey, CA 93943

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Defence Science and Technology Agency

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

The views expressed in this document are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol Number: N/A.

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release. Distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

Without corrective updates from the Global Positioning System, navigational capabilities are degraded significantly when the inertial navigation system becomes the only source of an unmanned aerial vehicle's movement estimate. Today, unmanned vehicles are easily equipped with a variety of passive sensors, such as video cameras, due to their increasingly lower prices and improvements in sensor resolution. The concept of using an image-matching technique on an input video camera stream was demonstrated earlier with real flight data using a single low-grade onboard sensor. This technique works by matching the stream of data from the camera with a pre-stored depository of geo-referenced reference images to estimate the current attitude and position of an unmanned aerial vehicle (UAV). Preliminary results indicated that unfiltered position estimates can be accurate to the order of roughly 100 meters when flying at two kilometers above the surface and unfiltered orientation estimates are accurate to within a few degrees. This thesis examines developed algorithms on a suite of video data, seeking to reduce the errors in estimating attitude and position of a UAV. The data sets collected at King City and Camp Roberts, California, are also studied to discover the effect of altitude, terrain pattern, elevation map, light conditions, age of reference data and other parameters on estimation. This thesis concludes that in the absence of other sources of navigational information, imagery from a camera is a viable option to provide positional information to a UAV.

14. SUBJECT TERMS

image-matching algorithm, GPS-denied environment, UAS, UAV

15. NUMBER OF PAGES 109

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2–89)
Prescribed by ANSI Std. 239–18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

TEST AND EVALUATION OF AN IMAGE-MATCHING NAVIGATION SYSTEM FOR A UAS OPERATING IN A GPS-DENIED ENVIRONMENT

Keng Siew Aloysius Han
Civilian, Defence Science and Technology Agency
M.A., University of Cambridge, 2010
MPhil, University of Cambridge, 2010
B.A., University of Cambridge, 2009

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 2017

Approved by: Oleg Yakimenko
Thesis Advisor

Ryan Decker, United States Army Armaments
Graduate School at Picatinny Arsenal
Co-Advisor

Ronald Giachetti
Chair, Department of Systems Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Without corrective updates from the Global Positioning System, navigational capabilities are degraded significantly when the inertial navigation system becomes the only source of an unmanned aerial vehicle's movement estimate. Today, unmanned vehicles are easily equipped with a variety of passive sensors, such as video cameras, due to their increasingly lower prices and improvements in sensor resolution. The concept of using an image-matching technique on an input video camera stream was demonstrated earlier with real flight data using a single low-grade onboard sensor. This technique works by matching the stream of data from the camera with a pre-stored depository of geo-referenced reference images to estimate the current attitude and position of an unmanned aerial vehicle (UAV). Preliminary results indicated that unfiltered position estimates can be accurate to the order of roughly 100 meters when flying at two kilometers above the surface and unfiltered orientation estimates are accurate to within a few degrees. This thesis examines developed algorithms on a suite of video data, seeking to reduce the errors in estimating attitude and position of a UAV. The data sets collected at King City and Camp Roberts, California, are also studied to discover the effect of altitude, terrain pattern, elevation map, light conditions, age of reference data and other parameters on estimation. This thesis concludes that in the absence of other sources of navigational information, imagery from a camera is a viable option to provide positional information to a UAV.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

AGL	Above Ground Level
ASTER	Advanced Spaceborne Thermal Emission and Reflection Radiometer
BRISK	Binary Robust invariant scalable keypoints
DEM	Digital Elevation Model
DOD	United States Department of Defense
EO	Electro Optics
GPS	Global Positioning System
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
IR	Infrared
MOP	Measure of Performance
MSAC	M-estimator SAmple Consensus
MSL	Mean Sea Level
IMMAT	Image-Matching
RANSAC	Random sample consensus
RIL	Reference Image Library
RVI	Reference View Image
SIFT	Scale-Invariant Feature Transform
SURF	Speeded-Up Robust Features

UAV Unmanned Aerial Vehicle

UTM Universal Transverse Mercator

Executive Summary

The U.S. Department of Defense's Unmanned Systems Integrated Roadmap FY2011-2036 [1] identified autonomous operations within a Global Positioning System (GPS)-denied environment as key area of research, and this thesis studies the use of image-matching techniques to provide positional information in such a situation. Navigation systems within unmanned vehicles today are largely reliant on updates from the GPS and, in more capable systems, on the inertial navigation system (INS) as well. Within a GPS-degraded or GPS-denied environment on Earth or other planets, navigational capabilities are degraded significantly because the INS becomes the only source of a vehicle's movement estimate. Numerous unmanned vehicles today can and often are easily equipped with other passive sensors such as video cameras, as these devices have increasingly lower prices and improved sensor resolution. Such alternative sources of information can be used to work out the movement of the vehicle with respect to the operating environment. In the instance of video cameras, vision-based techniques can be harnessed for use as a navigation aid. Specifically, image-matching techniques rely on the stream of data from the cameras and a pre-stored depository of geo-referenced reference images to estimate the current attitude and position of a drone in flight.

In a 2016 work by Yakimenko and Decker [2], the researchers tested the concept of image-matching navigation on two different platforms using a single low-grade onboard sensor. Their preliminary results indicated that unfiltered position estimates were accurate to the order of roughly 100 meters when flying at two kilometers above mean sea level while the unfiltered orientation estimates are accurate to within a few degrees. This thesis extends the work by studying the errors associated with the estimated attitude and terrain versus the actual recorded GPS position during data collection flights conducted at King City and Camp Roberts in California. Various parameters that can affect the image-matching navigation algorithm performance are also studied at different altitudes and in two different terrains.

Five major observations from the conducted evaluations are as follows.

1. The Image-Matching (IMMAT) approach relies on the feature-richness of both satel-

lite and onboard camera images. To this end, a typical satellite image provides a resolution of $0.5m^2$ per pixel regardless of the size of the ground footprint. The resolution of on-board camera depends on the field-of-view (FOV, or zoom setting), altitude, and attitude. The best resolution is achieved in a level straight flight at low altitudes with a maximum zoom in. Nevertheless, such a setting results in a very narrow field of view (significant reduction in the number of features that can be used to match those of the satellite image). Specifically, with the TASE-200 sensor used in this research and a field-of-view of 35 degrees (Camp Roberts' flights), a resolution of $0.5m^2$ per pixel can be achieved only when flying below 400m AGL. Likewise for King City flights, where the videos were taken at field-of-view of 10 degrees, only flights below 1200m can achieve $0.5m^2$ per pixel resolution.

2. The texture of the Earth's surface has a major role. Specifically, flying over the agricultural area consisting of crop fields (between Greenfield and King City) at low altitudes with a narrow field-of-view results in no features detected in the onboard camera field-of-view. Some features can be detected only when flying in between the crop fields. One way to mitigate this effect might be increasing the field-of-view, but that leads to a decrease in resolution and possible failure to find the matches between two different resolution images. Still, this approach is worth exploring in the future.
3. Onboard camera stabilization (i.e., suppression of vibrations) has a crucial role, as well. In this research two aerial vehicles were used. The same sensor, a TASE-200, had much better stabilization when flying on UAV at 25m/s compared to that of a manned Cessna-206 flying twice as fast.
4. Varying the terrain elevation also contributes to the accuracy of IMMAT navigational solution. That includes a requirement to have a detailed terrain elevation map of the intended area of operations.
5. Aircraft attitude plays a major role, as well. In this research, IMMAT performance was evaluated only for straight level flight. Future evaluation should consider IMMAT performance while turning, climbing and descending.

Using a limited set of test data based on a (not high-end) TASE-200 sensor with some vibration isolation problems along with incorrect reporting of pan-tilt information (which was discovered within this research effort and reported to the manufacturer) resulted in an unusually high drop rate. This occurred when there were not enough matching points to

construct a projective transformation, which is a basis of the IMMAT approach. Nevertheless, this thesis was able to conduct a detailed assessment of the overall performance of the IMMAT algorithm.

The main conclusion is that when all conditions are met (i.e., at least five matching points are found), the IMMAT algorithm can provide an estimate of an aerial vehicle's position that is accurate to within $50m$ from its true position (this value correlates with the satellite image resolution), and determine the vehicle's attitude within ± 15 degrees for pitch and roll, while finding its yaw angle within just ± 2-degree accuracy.

Some additional observations follow.

- For the same field of view, as the flight profile increases in altitude, allowing more of the local terrain to be captured, with a consequential increase in the number of features and the likelihood of matches, the drop rates for the IMMAT algorithm decreases.
- If an IMMAT drop does not occur, then the error associated with IMMAT estimation appears to decrease with the altitude or pixel-per-meter on the ground.
- This thesis relies on a simple two-dimensional projection of satellite imagery into the view of a would-be camera in flight. The lack of elevation data introduces perspective differences that may contribute to the errors in estimation by the IMMAT algorithm. To quantify the errors due to projection further, two experiments can be conducted. First, real video imagery can be taken at various tilt angles, with the most important being vertically downward. The downward view matches best with the top-down satellite view and also obviates the need for terrain elevation information for projection purposes. The second is to enhance the projection algorithm by capturing a view from a three-dimensional satellite-image textured digital elevation model from the perspective of the camera, and comparing the estimates with the current approach.
- While the Reference Image Library can be created from a large collage of high-resolution satellite images prior to flight and then stored onboard the UAV, it can require quite a bit of space to store the frames. For example, a nominal trajectory that requires about 700 reference frames stored in high resolution amounted to 0.5GB; storing only the extracted features and using only those will require much less space. This presents an opportunity to investigate a method for storing the Reference Images Library that can work with the IMMAT algorithm efficiently.

- As the IMMAT algorithm produces an estimate frame-by-frame and only when sufficient matches are found, there will be variations in the estimates generated when they are produced; otherwise, there are no estimates. The question is whether feeding the output of the IMMAT algorithm into a Kalman filtering process will (1) produce a cleaner output, (2) produce more accurate positional predictions, and (3) use the previously known positional predictions as input initial positional estimate into the six-degrees-of-freedom optimization procedure.

Overall, the work within this thesis enhances the users' understanding of deploying IMMAT algorithms for guided unmanned activities that may follow a predetermined trajectory. With a predetermined trajectory, recently captured high-resolution images of the operational environment that the planned trajectory is expected to fly over can be pre-loaded onto the unmanned system. In this way, it can be used as an alternative navigational aid when other on-board navigational equipment fails or cannot be used. One specific example of where the findings of this investigation are useful is in autonomous military operations within the GPS denied environment that render an external accurate means of navigation unavailable for unmanned navigation.

List of References

[1] Department of Defense. (2011). The Unmanned Systems Integrated Roadmap FY2011-2036. [Online]. Available: https://my.nps.edu/documents/106607930/106914584/UxV+DoD+Integrated+Roadmap+2011.pdf/0f123fb1-ef1f-4842-9855-85a136b28a93. Accessed September 13, 2017.

[2] O. Yakimenko and R. Decker, "On the development of an image matching navigation algorithm for aerial vehicles," *Proceedings of the IEEE Aerospace Conference, Big Sky, MT*, 2016.

Acknowledgments

I am grateful to many people for this work. First and foremost, I thank Professor Oleg Yakimenko for suggesting that this project would be appropriate to investigate, given that I would have preferred to work on something more relevant to my future job posting, and something more quantitative, involving some degree of coding. I thank him also for being a fantastic MATLAB aficionado and for having been the best lecturer for MATLAB I have had in the last decade. He has also helped significantly with my understanding the data sets that were used, as the logs from the systems used were not fully matching up with the flight profile due to system-setup calibration biases.

I would also like to acknowledge Assistant Professor Ryan Decker from the United States Army Armaments Graduate School at Picatinny Arsenal for providing the initial code-base upon which the rest of my work was built, and for ideas for enhancing the algorithms further. I gratefully thank all my proofreaders: Michele D'Ambrosio, Ms. Barbara Berlitz, Charles, Anna, and my mom. Proofreading is painstaking work, requiring many hours to suggest reorganization and rephrasing to better present my ideas. I am also happily acknowledging the financial support of my sponsor, the Defense Science and Technology Agency of Singapore. Without their support, the opportunity to pursue this master's program at the Naval Postgraduate School would not have materialized.

I also have to thank Renee, Michele, and Simone, a welcoming family who provided me with a comfortable living environment during this period when I was busily trying to finish this thesis. I will always remember this chapter of my life —learning how to cook authentic Italian cuisine and savoring the most amazing raviolis. I had not tried anything as amazing as the truffle ricotta ravioli up to that point in my life! I would also like to mention a very special friend, Charcoal, although he will not be able to read this, as he is the family dog, but he has kept me company after school every afternoon when I thought through my thesis.

Last but not least, it was Wee Leong and Jeremy who made this one year at NPS a more pleasant place; my experiences would have been vastly different without their support.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

This chapter provides the background, context and the setting for the exploration of image-matching algorithms for use as autonomous vehicle navigation aids. The main objective of this chapter is to formulate the problem statement, which is presented together with the motivation for this body of work.

1.1 Background

Most manned and unmanned vehicles flying today rely on an integrated Inertial Navigation System (INS) and Global Positioning System (GPS) navigation system that uses GPS to provide corrections to vehicle position at the rate of 1 to 10 Hz [1]. The GPS uses transmitted information from at least four satellites out of a constellation of 24+ satellites (see **Figure 1.1** to compute its location, see **Figure 1.2** for an illustration). The GPS signal can become unavailable due to various natural phenomena or by human action; when it does happen, it is broadly classified as “GPS denial" [2], [3].

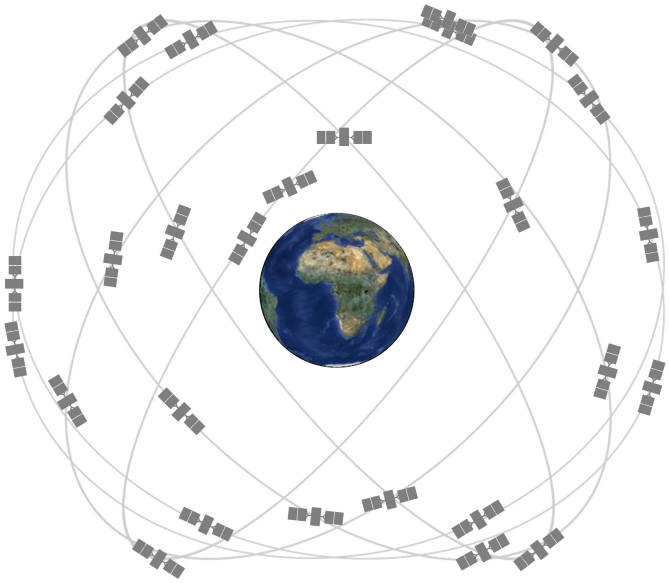

Figure 1.1. GPS constellation. Source: [4].

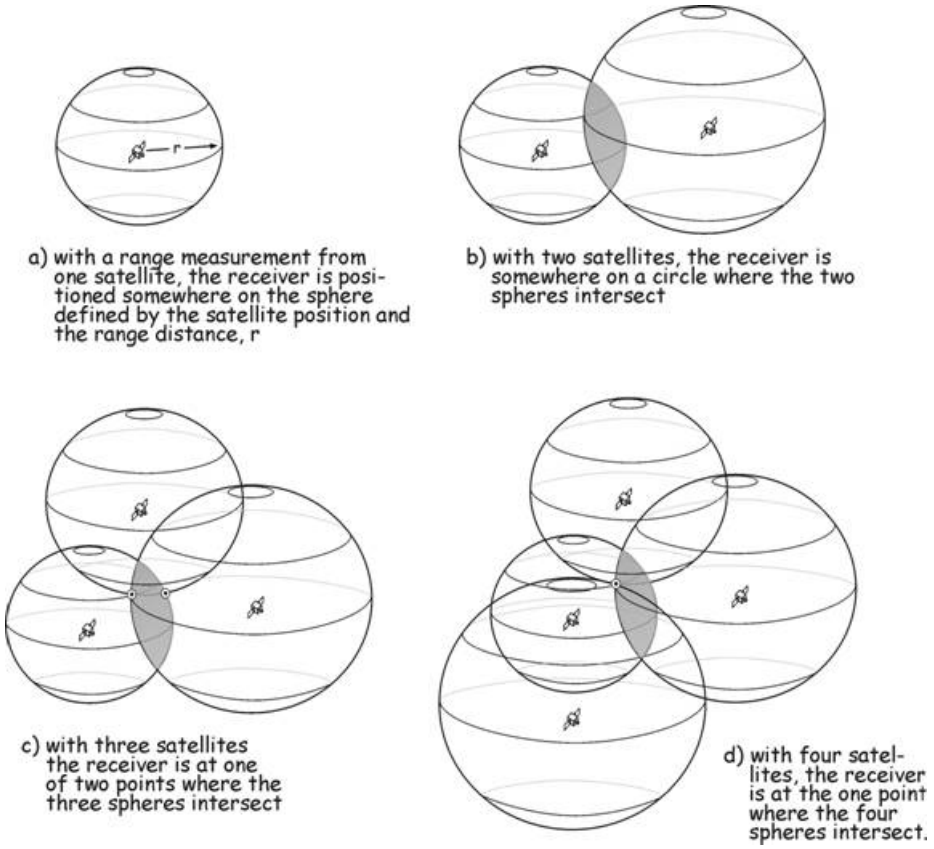

Figure 1.2. GPS triangulation. Source: [5].

Without GPS positional updates to calibrate the INS, navigational capabilities quickly degrade when the system relies solely on the INS to drive dead reckoning estimates. The question at hand is whether there are alternative mechanisms, preferably sensors already available, which can provide another source of positional feeds into the navigational system.

Numerous unmanned vehicles today can and are often easily equipped with other sensors. These alternative sources of information can be used to work out the movement of the vehicle with respect to the operating environment. One such sensor is the video camera; cameras are (1) getting increasingly cheaper, (2) improving in sensor resolution, and (3) getting smaller. As such, the use of a video camera as an alternative source of navigation information is the prime focus of investigation within this thesis.

In 2016, Yakimenko and Decker [3] demonstrated that the concept of image-matching (IMMAT) navigation shows promise with both simulated data and real flight data captured

from a single low-grade onboard sensor. Their study reported preliminary results that unfiltered position estimates are accurate to roughly 100 meters (m) when flying at two kilometers (km) above the Earth's surface and unfiltered orientation estimates are accurate to within a few degrees. Yet, further analysis is necessary to characterize the performance and behavior of the algorithm better.

1.2 Motivation and Problem Definition

This thesis seeks to further study the behavior of the proposed IMMAT concept. Understanding the behavior of algorithms allows users of the algorithm to achieve more robust performance during operations. Studies to reveal the effects of altitude, terrain pattern, elevation map and other parameters on IMMAT navigation algorithm performance can help users to better understand the promises and limitations of the IMMAT approach.

This thesis addresses the problem of testing and evaluation of an IMMAT algorithm using two sets of video data collected by manned and unmanned aerial vehicles equipped with a representative sensor.

The work within this thesis spans the domains of computer vision, systems engineering and unmanned aerial vehicle navigation. The broad intent of this investigation is to develop new techniques using onboard image stream or video, processing those images with the intention to characterize the motion of autonomous aerial vehicles so as to support navigational tasks.

This effort contributes towards autonomous operations within a GPS denied environment, and the objectives are aligned with the United States Department of Defense (DOD)'s Unmanned Systems Integrated Roadmap FY2011-2036 [6].

In order to better develop algorithm for the image-matching navigation task, this research conducted functional analysis [7] as guided by systems engineering best practices. This analysis enables us to gain greater insight into how to divide the task according to different algorithmic procedures. A high-level schematic of the Image Navigation task is depicted in **Figure 1.3**. The sub-functions are labeled individually, and a description of each is presented in **Table 1.1**. The functional decomposition helps subsequently by structuring the implementation of an IMMAT algorithm that is described in the rest of the thesis.

Table 1.1. Functional decomposition for the image-matching navigation task.

Label	Function	Description
F.0	Image Matching Navigation	Broadly, the Image-based Matching Navigation task is about making navigational decisions relying on reference images of an operating environment that have reliable location information tagged to it. From an unknown location, pictures or images of the area are taken and then compared with the available reference images.
F.1	Manage Ground Truth Reference Imagery	A means to manage a repository of methodically organized images is needed to facilitate the image-matching task efficiently. The library shall contain ground-truth information such as latitude and longitude (or other equivalent location referencing mechanism) of the scene. The library must be able to be updated with appropriate reference frames.
F.1.1	Retrieving Geo-Referenced Imagery	An appropriate external source of retrievable high-quality geo-referenced imagery is needed, appropriate for the area of operations. Geo-referencing information needs to contain latitude and longitude. Having additional information such as the elevation of the ground at that point can also be useful.
F.1.2	Create Reference Images Library	Using the geo-referenced imagery, the user needs a method to generate a number of reference image frames according to a planned trajectory, such that when the UAV flies over the planned path, the scenery can be matched with these references to derive the aerial position and pose.
F.2	Flight Trajectory Planning	To select and create appropriate reference frames to be stored for cross-referencing, a means of path planning is necessary. The planned path will provide critical information such as latitude, longitude and altitude of UAV, the camera view point of the on-board sensor, as well as the underlying terrain height.
F.2.1	Determine Start and End points of Flight	There shall be a means for defining the starting and ending points of a flight.
F.2.2	Determine Nominal Pose of Camera	There shall be a means for defining the nominal roll, pitch and yaw of the camera, that is, what the camera is looking at during the flight.
F.3	Estimating Position and Pose of Camera in-flight	This is the core function of the image-matching navigation task - to use nominal trajectory information together with the reference library and the incoming sensor stream to produce an estimation of the in-flight camera pose and location.
F.3.1	Matching Video Frames to Reference Images	This sub-function finds the mathematical transform that would map the incoming video frames to an appropriate geo-referenced image frame and in so doing can produce the first positional estimate for the camera.
F.3.2	Perform Optimization of Roll, Pitch, Yaw and Positional Estimates	After having the rough position of the camera, this sub-function works to reduce the amount of error within the initial estimate for all six degrees of freedom - that is the 3-dimensional position as well as the roll, pitch and yaw of the camera.

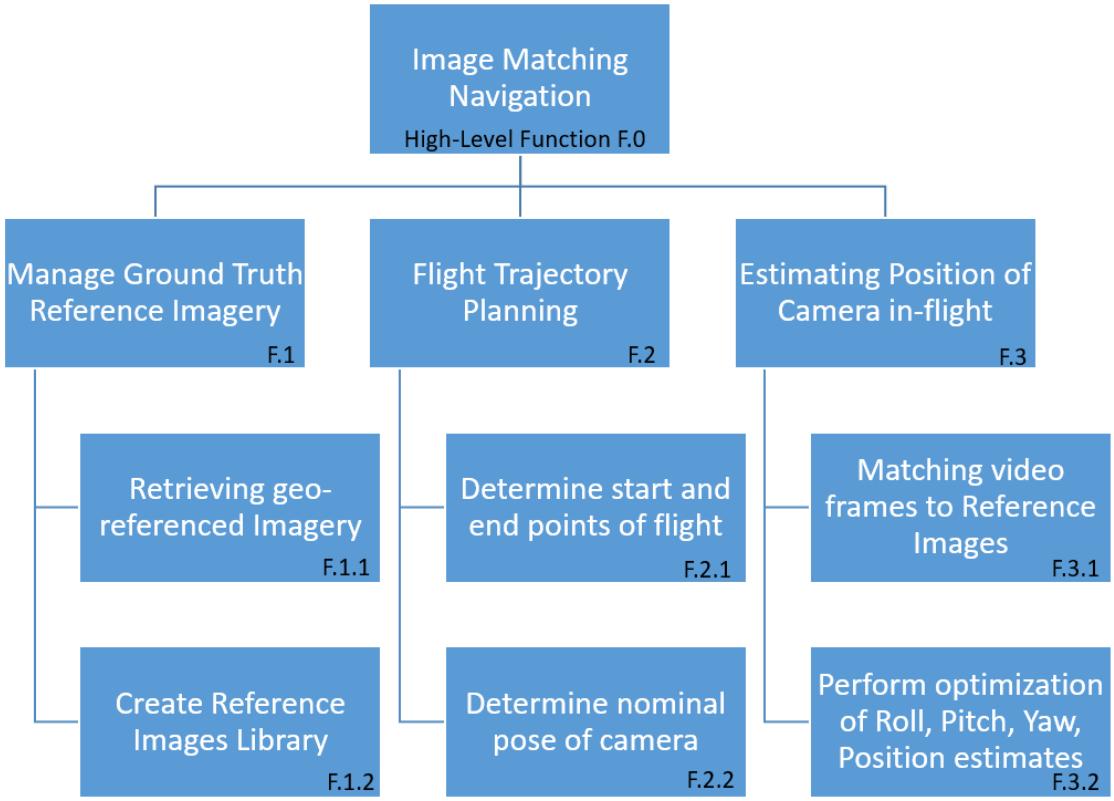

Figure 1.3. Image-matching navigation functional decomposition.

1.3 Organization of the Thesis

To address the problem formulated in Section 1.2, the remainder of this thesis is organized as follows,

Chapter 2 presents a review of existing literature documenting work previously done within the domain of the thesis. The chapter also summarizes relevant concepts such as the applicability of satellite imagery and digital elevation models, and the way these will be used to provide accurate geo-referenced images against which the environment and the data can be referenced and then modeled.

Chapter 3 presents the implementation details of the algorithms used for image matching.

Chapter 4 presents the datasets and data collection process used for this project. This chapter also provides a description of the physical system used to collect the flight data for analysis. The results are then analyzed and discussed.

Chapter 5 provides the concluding remarks about the research described in this thesis.

The wider implications of the results on future work and what research still remains to be done are also discussed.

Appendix A provides a full listing of all the meta-data that is made available by the camera used for this thesis.

Appendix B provides the meta-data details for the satellite imagery used for this thesis.

Appendix C provides a schematic and workflow of the MATLAB codes that were written for this thesis.

CHAPTER 2:
Relevant Concepts

This chapter presents various relevant concepts for the subject of this thesis. Concepts such as the reference frames for a UAV set against the world coordinates, image feature extraction algorithms, and the Kalman filter are also introduced.

2.1 Overview of Computer Vision Navigational Techniques

A large body of work is available pertaining to attitude estimation using various sensor inputs [8]. Sensors relied upon are variously the inertial navigation system, the on-board accelerometers, magnetometer, and most commonly today, the GPS. The focus of this thesis is on the use of the video stream that is available for most UAVs. The rest of this section presents a review of work done within the computer vision domain for attitude estimation of UAVs.

Mondragón et al. [11] proposed to use an omnidirectional sensor to identify a skyline and use it for attitude and heading estimation, noting that this system can be used as a redundant system for the INS and gyro-sensors. The omnidirectional sensor used in their research was a catadioptric video camera. **Figure 2.1** shows a hyperbolic reflector capturing an omnidirectional view of the surrounding environment; examples of the sensors themselves are shown in **Figure 2.2**.

Their approach requires the image contain the horizon-line from which their proposed algorithm segments the image to find the horizon. The detected skyline is then mathematically modeled as an occluding contour of the Earth as a plane inside a unit sphere, where the horizon forms a red line as the intersection of the plane of the Earth with the sphere (see Figure 2.3). The normal to the modeled plane provides a basis to estimate the pitch and roll. The yaw is estimated by checking registration of visual objects as they shift from frame to frame.

Kong et al. [2] used a feature-based navigation technique that essentially works by comparing

Figure 2.1. A hyperbolic reflector of a catadioptric sensor capturing an omnidirectional view of the surroundings. Source: [9].

features of an image with a previously taken set of reference images that are labeled with GPS data. The images taken by the onboard camera need to be mathematically transformed into the same plane as the reference images and then by feature matching. In their study, Kong et al. proposed using features that are as far as possible invariant under different lighting conditions. Their algorithm used edges extracted by the “Canny Edge Detector." The number of features extracted was deliberately kept small to reduce mismatch rates. A Gaussian blur filter was applied to reduce the number of unwanted features and smooth the edges extracted. To calculate the UAV’s position, the algorithm computed the centroid of a feature known to exist on the reference image (in world coordinates) and the image taken by the onboard camera. The motion could then be deduced by computing the translation vector between them. The authors concluded that *there are limitations on matching natural features*. Also, the authors proposed as a next step to accelerate the computation by moving it onto a Field Programmable Gate Array (FPGA) as the algorithm is floating point intensive

Figure 2.2. Examples of catadioptric sensors. Source: [10].

and highly repetitive, which can benefit greatly from hardware acceleration. As this thesis also works on matching natural terrain features, any limitations in natural feature matching will also be noted.

Yakimenko and Decker [3] proposed using high-resolution satellite images with IMMAT algorithms to tune the position and attitude of a UAV. The proposed approach utilized the IMMAT algorithm to match a camera position to a geo-referenced satellite image. Broadly described, the concept is to optimize the location estimate of the features of the real-flight image on the satellite image using a feature detection algorithm. Further details of this approach are given in the next section of this thesis, which extends the preliminary work previously done and described in the reviewed literature. This work promotes the understanding of the effect of operations at various altitudes, and where possible, to improve on the accuracy of the technique.

2.2 Satellite Imagery and Digital Elevation Map

For a source of geo-referenced imagery, the use of the DigitalGlobe satellite imagery is introduced. Then, as the satellite imagery does not contain elevation information. elevation information associated with the area of operations is supplemented with digital elevation map of the terrain from the Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) Global Digital Elevation Model database.

Figure 2.3. Catadioptric mathematical model. Source: [11].

2.2.1 Satellite Imagery

Satellite imagery geo-referenced to the latitude and longitude, has been used to provide ground truth. Yakimenko and Decker [3] earlier recommended the use of the geospatial data provided by DigitalGlobe as it was the most accurate library of the Earth. As such, for this thesis high-resolution satellite imagery was retrieved from the DigitalGlobe website [12] for both Camp Roberts and King City, California.

DigitalGlobe's geospatial big data (GBDX) platform provides access to 15 years' worth of geospatial data along with the tools and algorithms necessary to extract useful information from that repository.

The high resolution satellite images of the area of interest can be made by selecting the desired image layers and then creating a mash-up image. This image collage can then be downloaded as high-resolution tiles that can be stitched together to form a large contiguous image of the area of interest. Each pixel within these high-resolution tiles represents a

Table 2.1. Characteristics of the ASTER digital elevation model.

Tile Size	3601 x 3601 (1° x 1°)
Pixel Size	1 arc-second
Geographic Coordinate System	Geographic latitude and longitude
DEM Output Format	GeoTIFF, signed 16-bit, in units of vertical meters Referenced to the WGS84/EGM96 geoid
Special DN Values	-9999 for void pixels, and 0 for sea water body
Coverage	North 83° to South 83°, 22,702 tiles

half-meter by half-meter square on the ground. It is from the stitched high-resolution image that reference images will be created for the Reference Image Library. The details of the reference image creation are presented in Chapter 3.

2.2.2 Digital Elevation Map

High quality, geo-referenced terrain elevation data is required in order to model the effects of the underlying terrain.

For the purposes of this thesis, the terrain models of the operating areas were retrieved from the ASTER Global Digital Elevation Model (DEM) Version 2 database, hosted by the United States Geological Survey (USGS) (https://www.usgs.gov/). The data is open-source and publicly downloadable. A DEM is essentially gridded data where each square in the grid corresponds to a geographic location, holding a value that represents the elevation above mean sea level. In the case of the ASTER DEM, the data was stored as a gridded (latitude, longitude, elevation) matrix within a geoTIFF file.

To download the relevant digital elevation model, we entered the bounding latitudes and longitudes into the USGS EarthExplorer system (https://earthexplorer.usgs.gov/) and selected the ASTER database. The system then made available the appropriate data package for download. The data is retrieved as a geo-referenced TIFF file with 16-bit information of vertical meters, where each pixel represents 1 arc-second by 1 arc-second (approximately

30m by 30m near the equator) in geographic latitude and longitude. This data is captured by the National Aeronautics and Space Administration(NASA) Terra spacecraft's infra-red (IR) cameras with a 20-meter elevation accuracy at 95% confidence interval. The information within the DEM is used to set the elevation of the ground to provide for better re-projection for the creation of the Reference Image Library. Details of the DEM model used in this thesis are captured in **Table 2.1**.

2.3 Reference Conventions

For the algorithms to work, a consistent set of reference frames must be used to properly describe the orientation of an aircraft in three-dimensions around its own center-of-gravity, as well as for referencing its position within the world coordinates.

This section lays out the referencing conventions used within this thesis. The first part introduces the world coordinate reference frame, with which the unambiguous location of the UAV can be described. Following that, the convention for describing the attitude of the UAV is described.

2.3.1 UAV Body Frame

The UAV body frame of reference is body-fixed. It is fixed upon the center of gravity of the UAV. The convention used within this thesis has the $+Z$ pointing out of the bottom of the UAV, $+X$ out of the nose, and $+Y$ in the direction of the right wing, in other words, $x = north, y = east, z = down$ (See **Figure 2.4**, where the diagram depicts the world frame of reference in Latitude, Longitude and Up, in which targets and the platform physical location will be located with. In the air, the UAV is illustrated using a North-East-Down convention body-centered frame-fixed reference axes). Although it appears counter-intuitive to use a coordinate axes that is oriented differently to the world frames, the advantage of using this reference frame for the UAV allows for easier mathematical transformations when computing rotations and translations with respect to the ground.

2.3.2 Universal Transverse Mercator Coordinate System

The work within this thesis is primarily about estimating the location of a UAV with respect to the Earth's surface. To make this estimation, there is a need to unambiguously

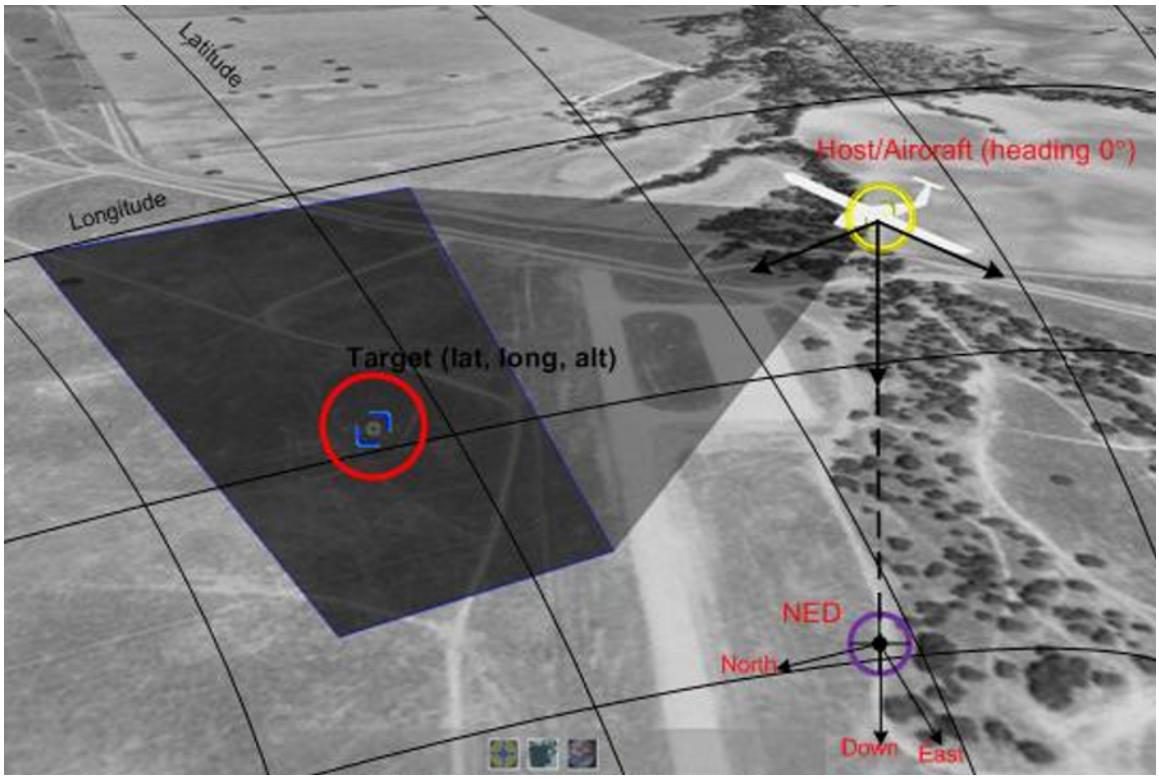

Figure 2.4. World and UAV frames of reference.

reference a location on the surface of the Earth. This thesis uses the Universal Transverse Mercator (UTM) Coordinate System to identify locations on the surface of the Earth as the units correspond to meters on the ground. This method greatly simplifies the computation of distances in three dimensions. Further, 3DEM provides the capability to convert any terrain using Geodetic (latitude-longitude) projection into a UTM projection. Terrain data sources such as the NASA SRTM data and the National Elevation Dataset are provided in a geodetic latitude-longitude projection. The disadvantage of the geodetic projection is that it introduces an east-west distortion at high latitudes. The UTM projection corrects this distortion, providing a more realistic map view and 3D scene. An added benefit is that using the UTM projection is helpful in the application of terrain overlays.

2.3.3 Image Features Extraction

For the purposes of this thesis, image features are data found within either the satellite images or the sensor video frames that are relevant to solving the proposed image-matching problem.

Many image feature extraction algorithms have been developed, for example, Speeded Up Robust Features (SURF), Binary Robust Invariant Scalable Keypoints (BRISK), Lowe's [13] Scale-Invariant Feature Transform (SIFT).

Although Lowe's SIFT algorithm is effective in situations where image features are invariant even when common image transformations are applied, the effectiveness comes at the expense of computational cost (i.e., it is slow) [14]. By contrast, SURF was described in 2006 by Bay et al. [15] and was demonstrated to be significantly faster than SIFT, and thus, suitable for the purposes of this thesis. Similarly, BRISK [14] is another plausible alternative that is rotation and scale invariant. It is also suitable for matching up feature sets that are likely to be the transformations of those image features, but that method is not explored within the scope of this thesis and is left for future work.

2.4 Random Sample Consensus Algorithm for Outliers

In this work, features from images are extracted and then a matching correspondence between the most features in two similar images is estimated. This matching may contain outliers and do not accurately describe how the features match up with each other in the two images. To exclude spurious matchings, the estimated correspondence ran through a Random Sample Consensus (RANSAC) algorithm. The RANSAC algorithm, first described in 1981 by Fischler et al. [16] seeks to find a consensus set of inliers that can best explain the match between two images. Briefly, the RANSAC algorithm steps through the following to produce a model to fit the data, assuming the model has a parameters vector $\vec{X}$:

1. Select a subset of N out of M data points at random
2. Hypothesis generation step: use the selected N points to estimate $\vec{X}$
3. Hypothesis verification step: count the number data points in M fits the model within a configurable tolerance. Call the proportion of data points fitting the model p.
4. if p is sufficiently good, exit RANSAC algorithm and flag success.
5. Otherwise, go back to step 1 and repeat for Q times.
6. Exit after Q trials, flag failure - unable to find a model that adequately explains the data.

This thesis uses a variant of the RANSAC algorithm, which is called the M-estimator SAmple and Consensus (MSAC) algorithm. The MSAC algorithm uses optimization to

speed up convergence; a detailed evaluation of all the RANSAC variants was conducted by Choi et al. in 1997 [17], where the differences in the variants are detailed.

2.5 Estimating System State Using Kalman Filters

If the IMMAT navigation approach were treated as a measurement process of a UAV's position and attitude within the environment, then the output would contain noise and have uncertainty within each observation. Further, there could be omissions from the output of the image-matching algorithm should inadequate matches be found. In such instances, one approach to infer parameters or system states of interest such as position and attitude from the jumpy output is the Kalman filter [18].

Broadly explained, a Kalman filter aims to minimize the mean square error of the parameters, assuming the noise in the measured data is Gaussian. Kalman filters are widely used in the military context to track targets by radar, for example. The Kalman filter is used to filter the outputs of the IMMAT algorithm to suppress the Gaussian noise.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3:
Image-Matching Paradigm

This chapter provides IMMAT implementation details within the MATLAB environment to take advantage of image processing toolkits, efficient matrix-based operations and the inbuilt-optimization algorithms. The algorithm developed will be used to estimate the UAV's position and attitude.

3.1 IMMAT System Architecture

The overall IMMAT navigation concept as proposed by Yakimenko and Decker [3] is presented graphically in **Figure 3.1**. As depicted, the IMMAT task is executed in several stages. The workflow as illustrated in the diagram is elaborated in the ensuing sections, and it matches up with the functional decomposition that was conducted in Chapter 1. The image-matching task is executed in several stages. Broadly, the concept steps through two main phases: the planning phase and the real-time operations phase. The planning phase contains all the steps leading up to the generation of the Reference Image Library, while the real-time operations phase contains all the steps after.

3.2 Generating the Reference Image Library

All steps up to and including the generation of the Reference Image Library are done in the planning phase. The planning phase involves tasks and activities that can be done ahead of time, preferably off-line, in preparation of the real-time phase. Some stages can be performed off-line (which in the case of a UAV, means pre-flight) as those tasks can be planned and prepared ahead of time, and do not require real-time processing on-board the UAV. One such step that is suitably performed off-line is the (1) planning of an anticipated trajectory of the UAV and then (2) generation of the Reference Image Library (RIL), which will be used by the UAV in-flight. For this phase, there is a need for an *a-priori* nominal trajectory, which is a limitation of this approach.

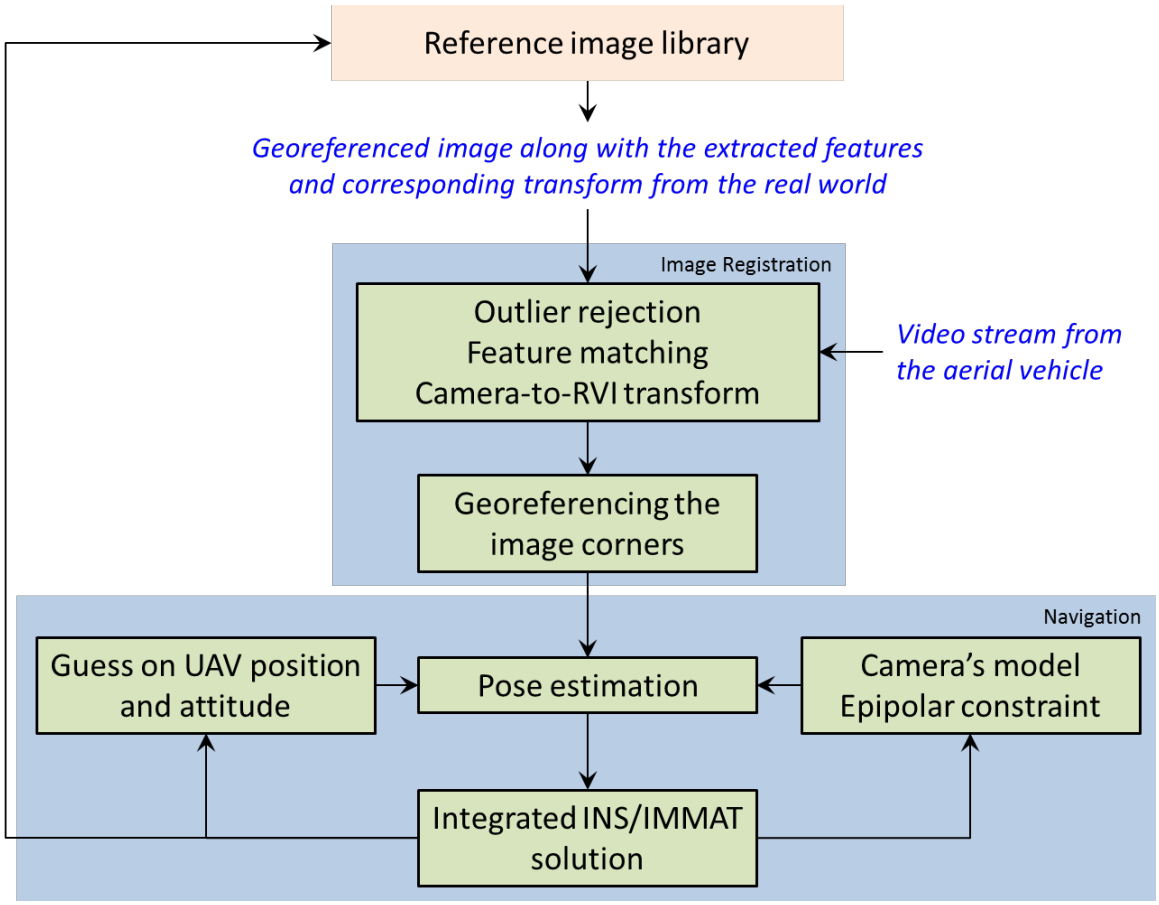

Figure 3.1. Schematic of the Image-matching algorithm workflow. Source: [3]

3.2.1 Planning Nominal Trajectory

Assuming that the UAV has been assigned a mission within a known area of operations, an operator can roughly plan a trajectory the UAV is expected to follow. This planned trajectory is termed the *nominal trajectory* for the flight.

For the purposes of this thesis, the nominal trajectories used for the research were created from real flight profiles (that are presented in detail within the next chapter) as the nominal trajectories have accompanying ground truth information available for further analysis.

The nominal trajectory for this research was created from raw flight data. First, a 25-period running average of the data points was used to address aliasing effects due to repeated data points. Then to smooth the planned trajectory, we fit a polynomial to the data. This is

described by the following block of MATLAB pseudo-code:

```
NSmooth=25;
NominalTrajectory.t=smooth(RawTrajectory.t,NSmooth);
NominalTrajectory.East_m=smooth(RawTrajectory.East_m,NSmooth);
NominalTrajectory.North_m=smooth(RawTrajectory.North_m,NSmooth);
NominalTrajectory.Lat=smooth(RawTrajectory.Lat,NSmooth);
NominalTrajectory.Lon=smooth(RawTrajectory.Lon,NSmooth);

NominalTrajectory.Roll_deg=smooth(RawTrajectory.Roll_deg,NSmooth);
NominalTrajectory.Pitch_deg=smooth(RawTrajectory.Pitch_deg,NSmooth);
NominalTrajectory.Yaw_deg=smooth(RawTrajectory.Yaw_deg,NSmooth);
```

The in-flight phase (which is described in detail in **Section 3.3**) relies on a repository of images against which an IMMAT algorithm compares incoming video frames from the onboard UAV sensor to estimate the pose of the vehicle. To build this Reference Image Library (RIL), satellite imagery of the known area of operations is retrieved, and then geo-referenced in the UTM coordinate system (which was discussed previously in **Section 2.3.2**).

Section 2.2 presented on reference sources of geo-referenced imagery. This section details how a reference image is created from the notional position and attitude of a Unmanned Aerial Vehicle (UAV) following the planned nominal trajectory.

The nominal trajectory as described in the previous section (**section 3.2.1**) is divided into N points. At each of those points, a series of high resolution images is extracted from the satellite images along the planned path the UAV is expected to take. Extraction is accomplished by using the nominal camera pose at those positions used to generate the Reference Images. **Figure 3.2** shows a nominal trajectory superimposed on the raw track data that was collected from an actual UAV flight (actual flight collection is presented in Chapter 4). The nominal trajectory is divided into 35 segments in this example, where at each of those points a reference image will be generated. The pseudo-code used to generate positions for the reference images follows:

```
    timeVector = [1:numel(range)]';
```

Figure 3.2. Creating a nominal trajectory.

```
trajectory.X_fit = fit(timeVector, trajectory.XY_raw(:,1),'poly9');
trajectory.Y_fit = fit(timeVector, trajectory.XY_raw(:,2),'poly9');
trajectory.XY_fitted =
    [trajectory.X_fit(timeVector),trajectory.Y_fit(timeVector)];

refImagesTime=linspace(0,numel(range),35+1);
refImagesTime = refImagesTime(1:35);

trajectory.RefImagesXYPosition =
    [trajectory.X_fit(refImagesTime),trajectory.Y_fit(refImagesTime)];
```

After generating the positions for the reference images, the roll of the camera is set to zero, while the pitch and yaw follows those of the nominal trajectory. With the three-dimensional position, roll, pitch and yaw of the an imaginary camera, the four corners of the field-of-view of the camera is projected from that position to the ground plane. Where the projection intersects with the ground is a trapezium patch which will be cropped and warped into the camera's view.

3.2.2 Creating a Reference Image

With the position and attitude information of a camera following the nominal trajectory, the center-point of the camera's viewpoint is projected to the ground map, along with the four corners of the camera's field-of-view. The high-resolution map is then cropped to the area enclosed by the four corners and projectively transformed into a rectangular view. This represents a notional scene of what an onboard camera might see during a fly-pass (**Figure 3.3**). It is essential that the four corners of the camera view can be projected onto the ground

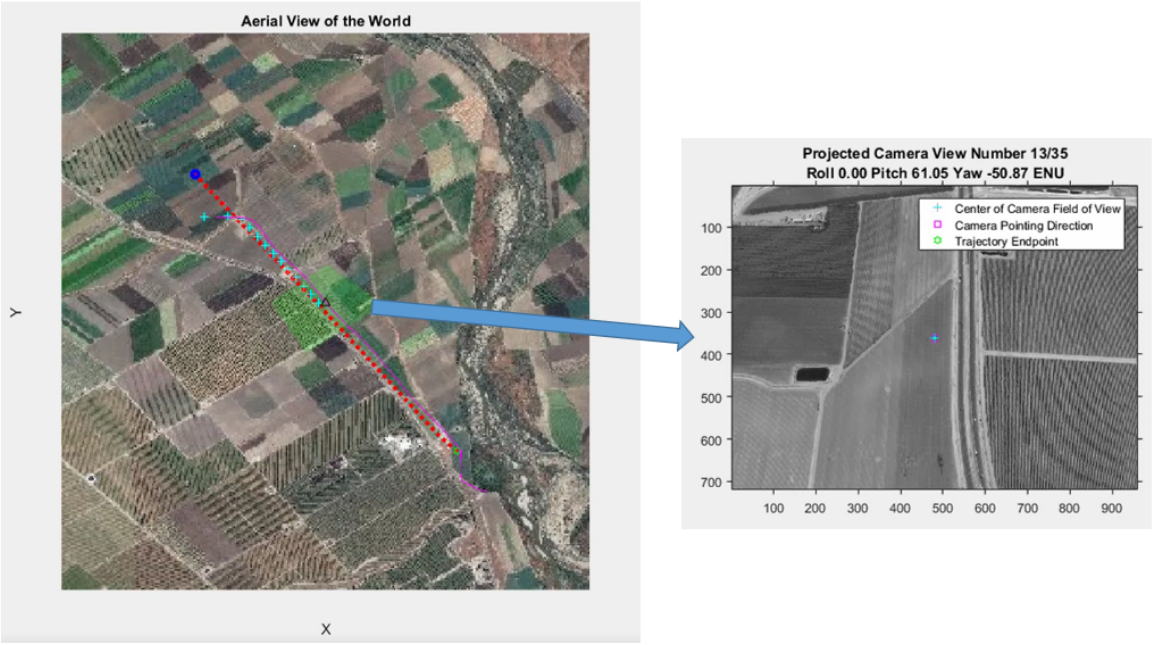

Figure 3.3. Creating a Reference Frame.

and not contain the horizon for this algorithm to work due to the basic, two-dimensional reference image generation scheme employed at this time.

During this stage when the reference image is created, the projective transform B, $\mathbf{A}_{3\times3}$ for mapping a pixel within the reference image (I_{ref}) to the real world coordinates (I_{ref}^{UTM}) is computed and stored with the reference image in the Reference Image Library. The equation that relates the image (u, v) pixel to the real world coordinates (x_{UTM}, y_{UTM}) is provided by

$$\begin{aligned} I_{ref}^{UTM} &= \mathbf{A}_{3\times3} I_{ref} \\ \begin{bmatrix} x_{UTM} \\ y_{UTM} \\ w \end{bmatrix} &= \mathbf{A}_{3\times3} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \end{aligned} \tag{3.1}$$

where $\mathbf{A}_{3\times3}$ is the transformation matrix and w is the scaling variable.

The generated reference images are stored together with the location and view perspective of the camera as the geo-referencing information within the RIL.

To further take advantage of the pre-planning phase, computationally hungry image-processing tasks such as feature extraction can be conducted on the reference view images (RVIs) and then storing the extracted features with the RVIs in the RIL before loading it on the UAV. This reduces the number of computational cycles onboard the UAV.

The implicit assumptions for the algorithm to work are the following:

1. The terrain as viewed from the UAV's camera can be adequately represented with a re-projected satellite view of the terrain;
2. Sufficient feature matches must be found between the RVI and the camera image to establish the transform between the two images;
3. The images coming through the sensor need to be downward-looking, so that the four corners of the sensor's field-of-view always intersect the horizon plane. The algorithm will fail as long as any one of the corners is projected above the horizon.

Once the RIL is generated, it should be stored onboard the UAV prior to mission deployment so that pose estimates during actual flight can be obtained.

3.3 In-Flight Phase

After the RIL is created, it can then be loaded onto the UAV so that it is available for in-flight use. The rest of this section describes how the reference frames within the RIL are used by an image-matching algorithm to estimate the location and the pose of the UAV.

The overarching idea for the image-matching algorithm is in two main stages. The first stage is to find a geometric transformation that is able to relate all pixels in the TASE camera's image to their geographic location. The second stage is to estimate the location and attitude of a would-be camera in space that would be able to create such a footprint of the features found in the TASE image on the ground.

3.3.1 Finding Matching Features in Reference Image and Camera Image

The "closest" corresponding reference frame within the RIL is selected by using information that may be available such as the last known coordinates of the UAV and then extrapolated by time along the heading it was previously taking.

After the appropriate reference frame has been selected, the features for that reference frame are matched with the features extracted from the onboard sensor image.

When attempting to match the reference image to the camera image, one may encounter three possible outcomes: (1) the scenes overlap and there are sufficient matches between the reference frame and camera image; (2) although the scenes match, there are insufficient matches between the reference frame and the camera image, and (3) no match is found between the reference frame and the camera image because the scenes within the images do not overlap. These three possibilities are elaborated each in turn, with graphical examples provided. Known situations where IMMAT drops might occur due to (2) and (3) are also be highlighted.

Sufficient Matches Between Reference Frame and Camera Image In this situation, the nominal trajectory provided usefully accurate position and pose for the reference image generation algorithm to capture the appropriate scene having a view that overlaps with the camera image. Furthermore, the reference image and the camera image are sufficiently feature rich that after the MSAC algorithm is run to remove outliers, there are sufficient

matching points for estimating a mapping transformation that maps camera image pixel position to real world coordinates.

Insufficient Matches Between Reference Frame and Camera Image In this situation, although the nominal trajectory provided a sufficiently accurate position and pose for a camera to generate a reference frame that overlaps in view with the camera image, there are inadequate inlier matches between the reference frame and the camera image to produce an estimate for the transformation that maps the camera image pixels to real world coordinates. In this situation, it would constitute an IMMAT drop. One example where insufficient correspondences were found is given in **Figure 3.4**. In that figure, while the Reference Frame scene matches the camera image, there are insufficient correspondences found after MSAC. For the top pair, green markers show the top few strongest SURF features that were extracted. The bottom pair shows the remaining features after RANSAC outlier culling, which is inadequate for estimating a geometric transform.. Another example attributable to a different reason where correspondences cannot be found is given in **Figure 3.5**. In this instance, the camera was experiencing vibrations and therefore insufficient inlier correspondences could be found between the reference images and the camera images.

No Matches between Reference Frame and Camera Image In this situation, the nominal trajectory location and pose used to generate the reference image do not overlap with the camera image. This could be due to perturbations in the flight profile or deviations from the flight profile that caused the camera not to view a scene that was expected to be viewed. This situation, like the previous case, would constitute an IMMAT drop. An example of this is found in **Figure 3.6**. For this example, the camera image was unable to match with the Reference Image as the IMMAT procedure had just switched to using the next Reference Image according to the nominal trajectory location prediction causing a mismatch between the scenes.

3.3.2 Finding UTM Coordinates of Matched Features by Optimization

Information available at this stage of the problem is (1) the live camera video frame (2) an appropriately selected reference image from the RIL, and (3) the parameters of the camera, such as the FOV and focal lengths in both horizontal and vertical directions.

The reference frame from the RIL is accompanied by geo-referencing information. The

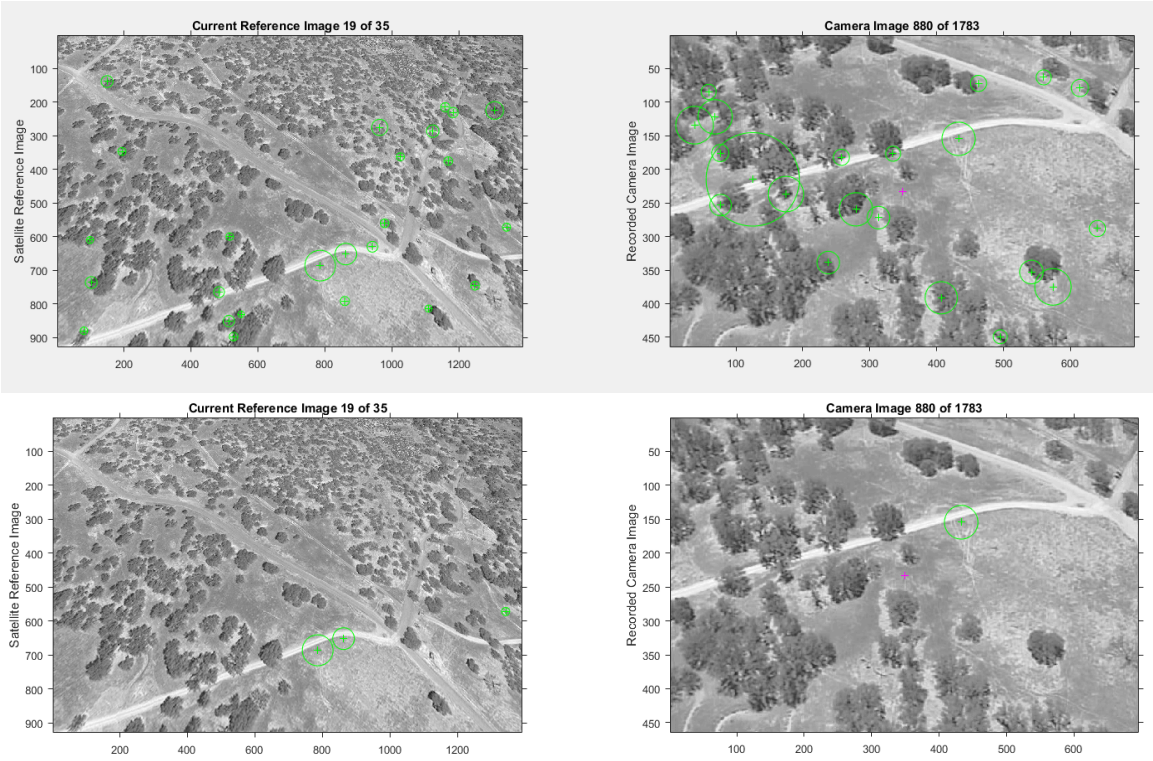

Figure 3.4. Insufficient matches.

reference frame itself had previously been warped based on the information available in the nominal trajectory, that is, the three-dimensional spatial position of the drone, as well as the viewing direction of the camera.

Assuming that sufficient features were found in the previous stage from extracting SURF features in both reference frame and the camera image (see **Figure 3.7**), these features will need to be paired in the next step.

An MSAC algorithm is used to sift through the features and find the best pairings between the features of both the video frame and the reference frame, discarding pairings that fall below a user configurable threshold. With the pairings, a two-dimensional projective transformation can be computed that maps the video frame view to the reference frame perspective.

Using the matched inliers between the reference frame and the camera image, a perspective transform is computed between the onboard sensor's $X - Y$ coordinates and the reference

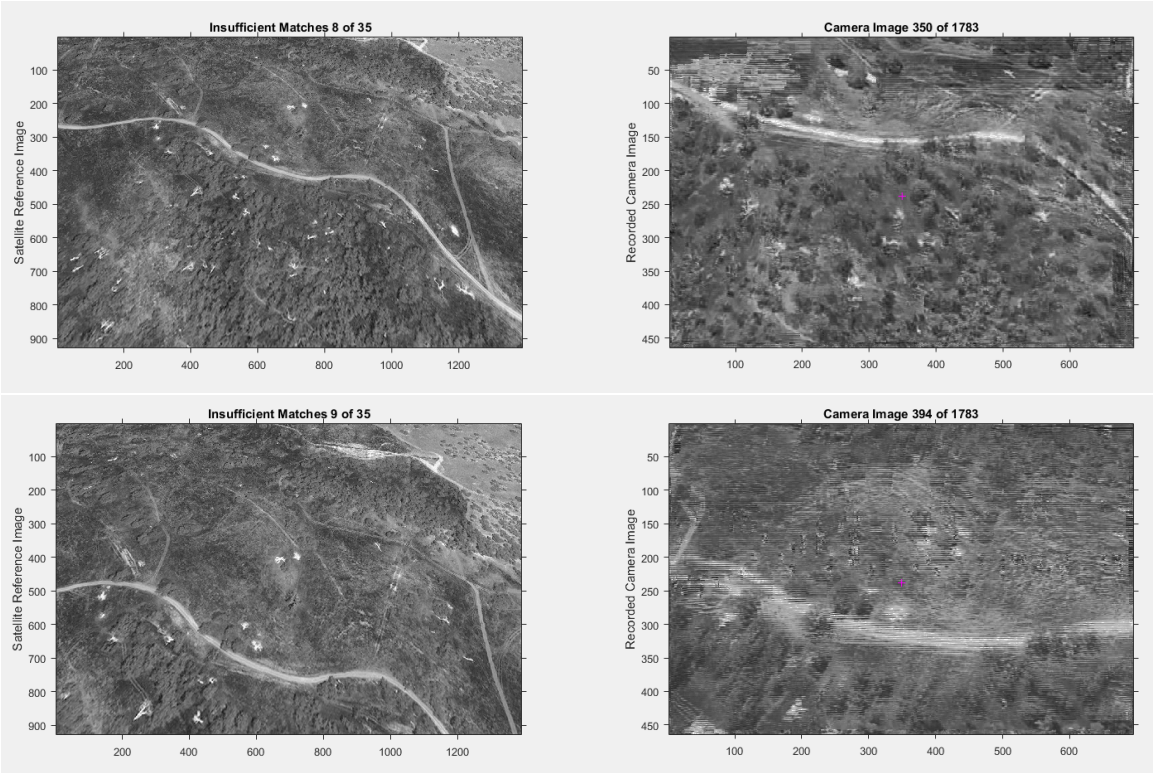

Figure 3.5. Insufficient inlier matches due to vibration.

frame's coordinates.

The matched pairs of features are then used to estimate a projective transform $\mathbf{B}_{3\times3}$ that maps features in the reference image (I_{ref}) to the features in the camera image (I^{cam}).

$$I^{cam} = \mathbf{B}_{3\times3} I_{ref} \tag{3.2}$$

Accounting for **Equation 3.1**,

$$I^{UTM} = \mathbf{A}_{3\times3} \mathbf{B}_{3\times3}^{-1} I^{cam} \tag{3.3}$$

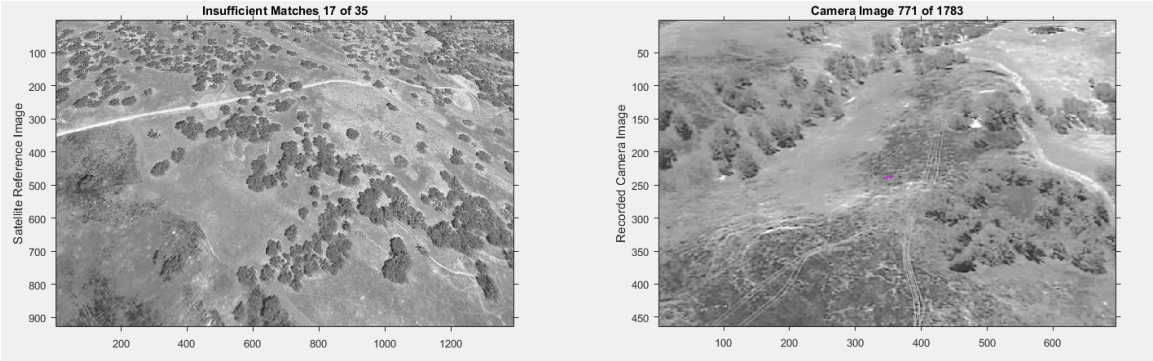

Figure 3.6. Example where the Reference Frame scene does not match with Camera Image.

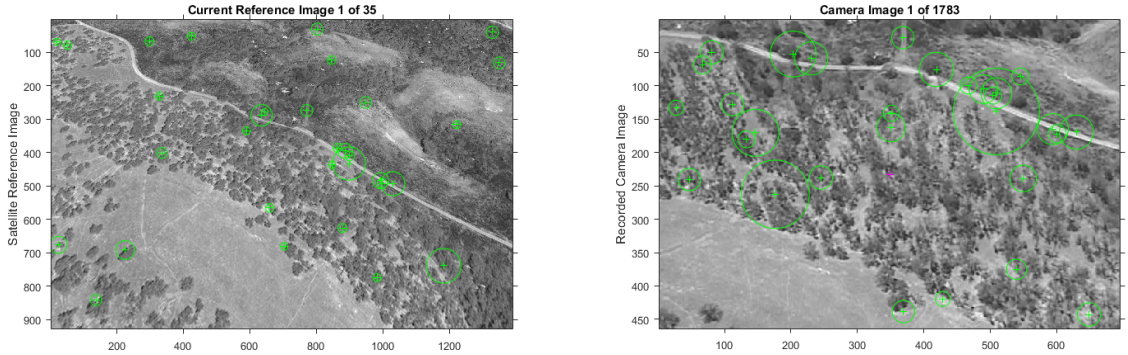

Figure 3.7. A coarse correspondence is found between features in the Reference Frame and Camera Image before MSAC outlier exclusion.

As the initial matching may contain outliers, it is run through the MSAC algorithm to remove outliers. **Figure 3.8** shows the corresponding features after outliers had been culled by the MSAC algorithm.

Until this stage, (1) the SURF features for both the appropriate Reference Image and the camera image are extracted, (2) a rough correspondence match between the two images are found, and then (3) the outliers in the matching are removed by the MSAC algorithm. **Figure 3.9** shows the result of a sample full run of a trajectory through the feature extraction and outlier culling process. The corresponding reference image and camera image each starts off numerous features (shown in the top sub-plot), which after an coarse matching significantly reduces to the order of tens (magenta plot). After the RANSAC/MSAC procedure for this example, on the order of about 5 –10 points are left which are used in the next stage to estimate the projective transform to find the UTM coordinates of those features.

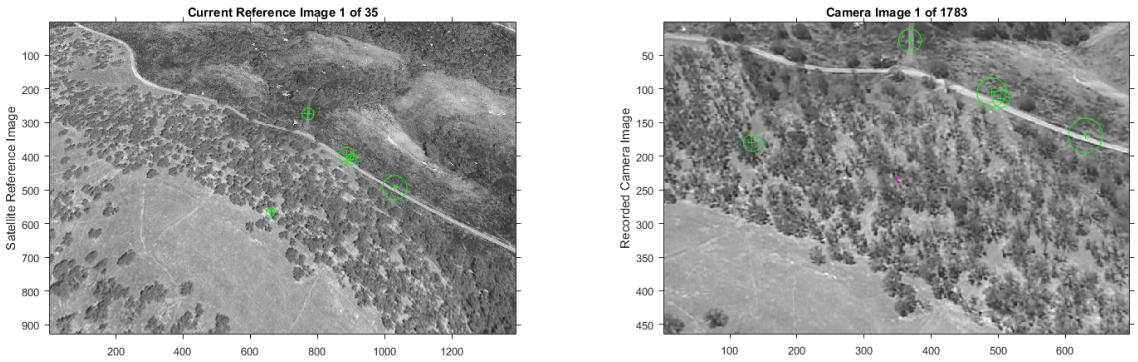

Figure 3.8. A montage of reference and camera image after MSAC outlier culling.

It is essential that the camera must be downward-facing so that the four corners of the camera's field of view can be projected onto the horizon plane. This approach fails when one of the projected corners lie above the horizon plane.

Using that relationship (transform), it is possible to project the matched points onto the ground. An optimization procedure is then executed to minimize the errors related to where the camera would have been in order to observe the points projected onto the ground in that way. In so doing, the hypothesis is that it is able to provide a reasonable estimate of where the camera was (position and orientation) when the image was recorded.

Estimating the Position and Attitude of the Camera. In the previous phase, actual geographical locations were identified for features within the TASE camera image. In this phase, the question at hand is to estimate the position and the attitude of the camera that would best match the same projected view on the ground. In other words, what are the best estimates that can be found from an estimated position and attitude in space for a would-be camera to allow the features of the camera frame to be projected onto the ground in UTM coordinates that maximizes the overlap with those points.

In diagrammatic form, **Figure 3.10** shows the UTM coordinate positions of four features (illustrated for simplicity as the four corners of the image) that were established in the earlier phase. From an estimated position and the attitude of the UAV, the features inside the camera frame as extracted earlier are projected onto the ground generating $guess_i$ positions, where i indexes each matched feature. These projected features should at this point of the procedure

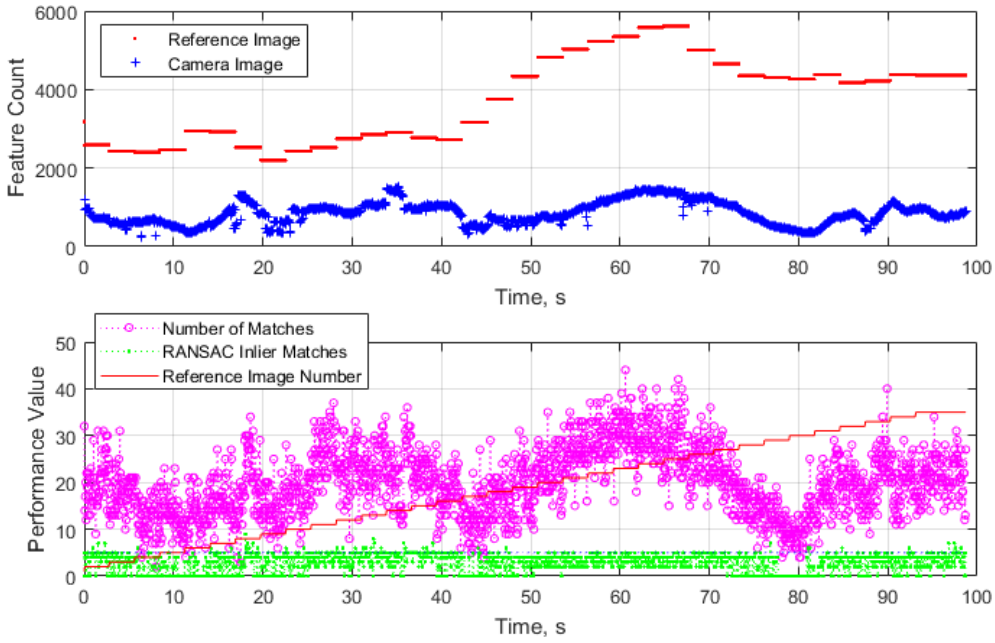

Figure 3.9. Feature extraction and outlier culling.

be close to the observed positions of the features.

The deviations are given by

$$\mathbf{\Delta}_i = features.position_i - guess.position_i \tag{3.4}$$

The cost function for this problem is defined to be the sum of squared deviations

$$ErrorSumOfSquares = \sum_i \|\mathbf{\Delta}_i\|^2 \tag{3.5}$$

Minimizing the error as computed by the cost function will produce the best estimate (within a configurable tolerance) for the location and attitude of the UAV.The reduction in the error between the observed value and the estimated value is done by an optimization algorithm.

During the concept exploration phase, an earlier implementation of the estimation process used an unconstrained optimization algorithm on the estimate of the UAV's position and

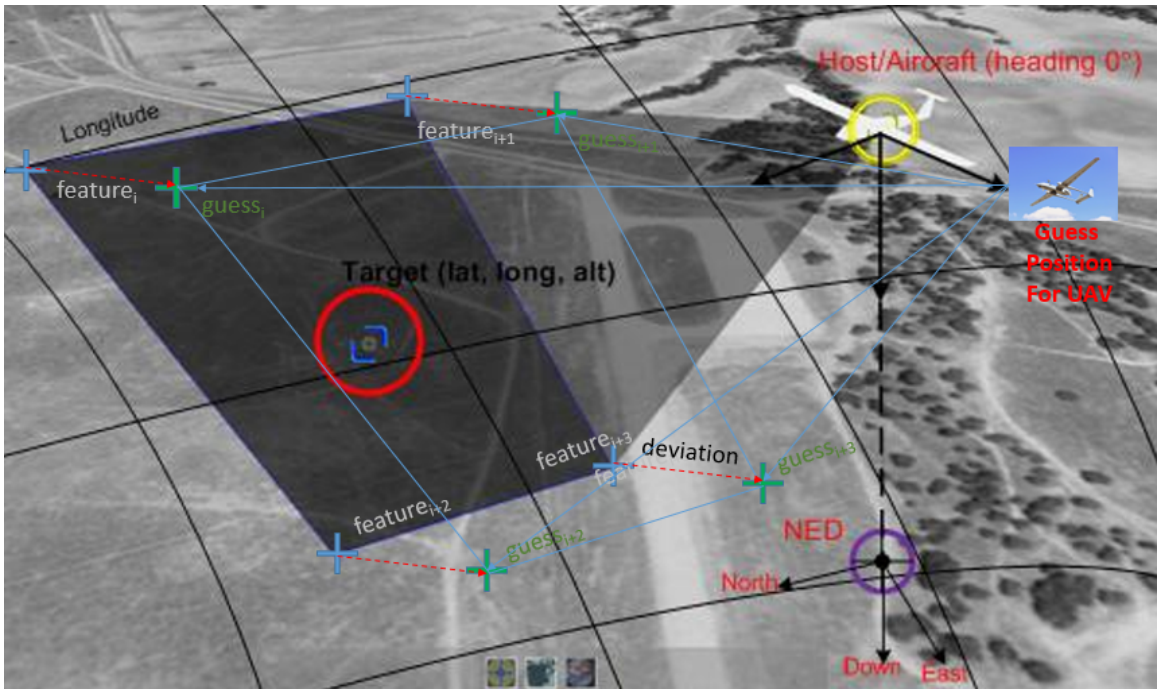

Figure 3.10. Projecting camera frame features (illustrated as corners for simplicity) onto ground in UTM coordinates from an initial estimated state [Easting, Northing, Up, Roll, Pitch, Yaw].

attitude. As a lot of information is known about the possible pose and location of a UAV given a planned trajectory, this information should be able to constrain the possible position of the would-be camera in space. In order to evaluate the efficiency and accuracy of constrained versus unconstrained optimization for the purposes of estimating the attitude and pose of a camera, a script was written to project the four corners of a viewfinder to the ground at a known position and pose (**Figure 3.11**).

For both the constrained and unconstrained search, the same initial estimate for position and pose of the camera were used. In the constrained search, the search was bounded to within 500m accuracy for position, and 30°, which approximates the bounds that will be set using interpolated position of the UAV in air as well as nominal roll, pitch and yaw of the UAV at that point in time based on the nominal trajectory.

A sample run from within the MATLAB environment using `fminsearch` and then the same problem done with `fmincon` is shown in **Figure 3.12**. The unconstrained search for an optimum took significantly longer to converge, taking close to 800 iterations versus 29

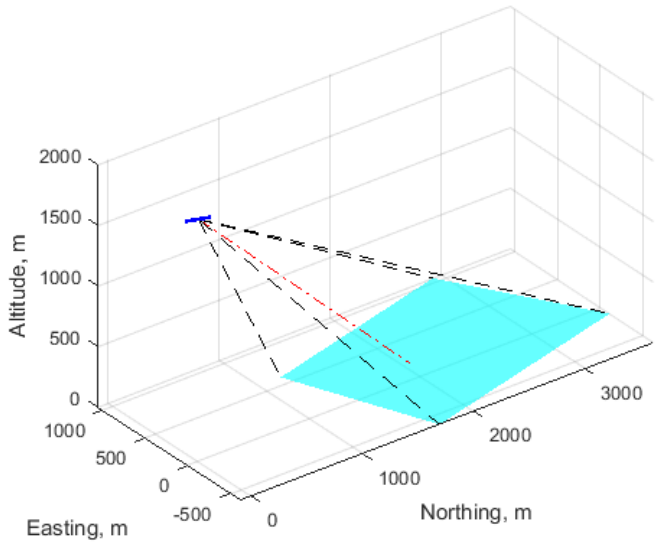

Figure 3.11. Viewfinder corner projection to ground.

iterations for the constrained search. The unconstrained search ended at a final error function value that is higher (i.e., worse) when compared to the constrained optimization solution (error function for unconstrained search was 226 versus computer zero for constrained search). Further runs confirmed that the constrained search found the solutions much faster and more accurately.

For the IMMAT algorithm, the constrained optimization algorithm is initialized with the orientation and the extrapolated position of where the UAV might have been if it were following the nominal trajectory. The boundaries were set within lower and upper bounds of nominal value ±30° for roll, pitch and yaw, and within nominal value ±500m for Easting and Northing. Boundaries were set for altitude nominal value ±500m and 50m the lowest flying trajectory which was 150m.

3.3.3 Pitfalls of Not Having Good Data

For the IMMAT algorithm to successfully estimate a projective transformation between the reference image and the camera image, a minimum of five control-point pairs are required. Using that as a filtering criteria where an estimate is considered valid only when there are five or more control-point pairs and running the algorithm on a representative trajectory

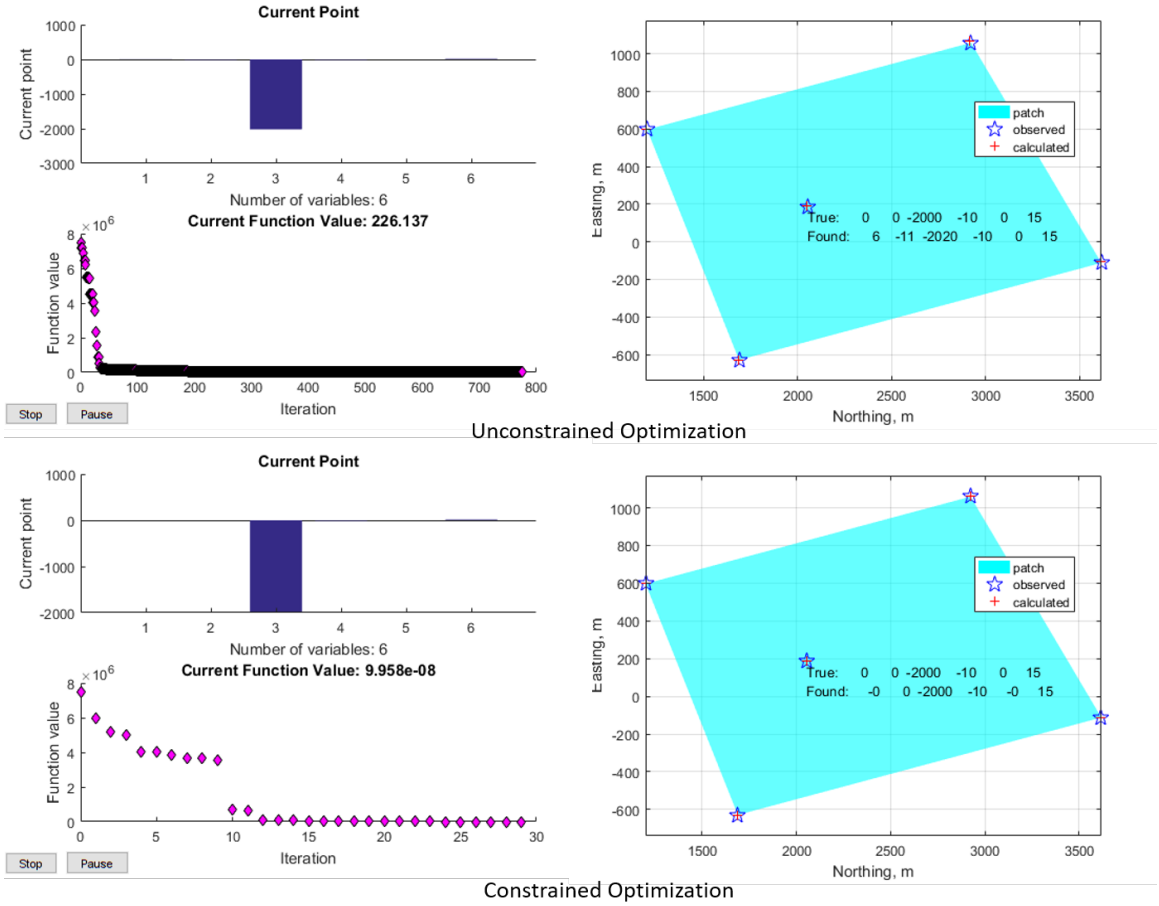

Figure 3.12. Unconstrained versus constrained optimization for estimating UAV position and attitude.

(as an example, taking one from Camp Roberts), then the observed drop rates are relatively high. **Figure 3.13** shows the output of a trajectory through the IMMAT algorithm.

For Camp Roberts, it was observed that even though the number of features extracted from the reference image and the camera image numbered in the thousands, after coarse correspondence (see the magenta plot in the second block of Figure 3.13), the number of matches drops significantly to the order of tens. After RANSAC is performed, there is a further reduction, with few remaining points that meet the five-or-more requirement for estimating a projective transformation. **Figure 3.14** shows the distribution of control points that was generated by the IMMAT algorithm, and **Figure 3.15** presents the data as a cumulative distribution. As can be seen, about 20% of the entire trajectory produces

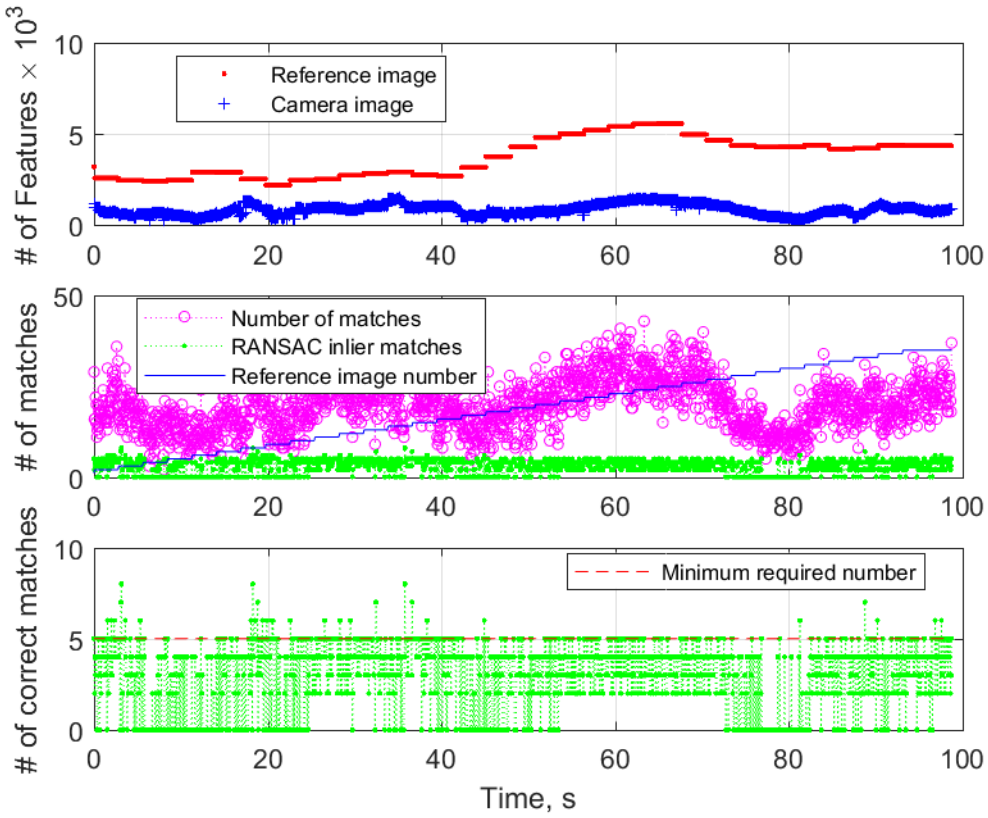

Figure 3.13. Sample output for a typical trajectory.

sufficient control points that can then be used to estimate a projective transformation.

Even though the number of data points that can be used to estimate the location and attitude of a UAV is not high, it is still possible to produce useful estimates by feeding the outcomes of the IMMAT algorithm to a Kalman filter (presented in the following section) which has built-in predictive capabilities. To this end, **Tables 3.1** and **3.2** shows two sample outputs for the IMMAT estimates for location and attitude as an illustration of when there are adequate matching pairs versus when there are insufficient matching pairs. As seen by looking at the error columns, when there are sufficient matches found, the IMMAT algorithm performs relatively well. By contrast, when the number of matches is below five and the project transformation cannot be computed, there is significant degradation of performance in the IMMAT estimates.

Apart from studying the magnitude of errors during the estimation phase, the effect on

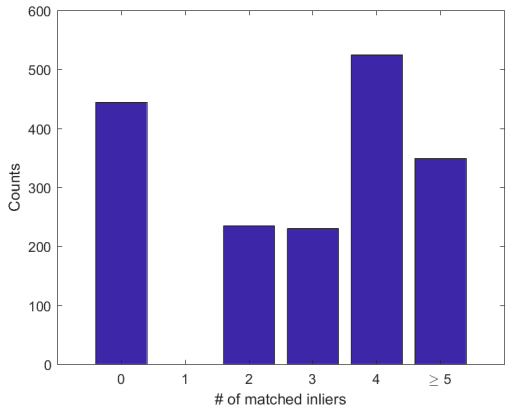

Figure 3.14. Distribution of control-point pairs.

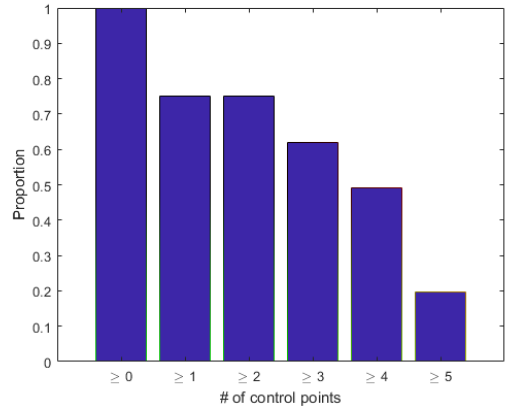

Figure 3.15. Cumulative histogram of control-point pairs.

convergence produced by sufficient versus insufficient control points was also studied. The rate of convergence for a seven control-point match is shown in Figure 3.16, while the rate of convergence for two matches is shown in Figure 3.17. With seven points, it was possible to compute a projective transform, yielding useful estimates that were able to converge, giving a final cost function value of around 11. In the other case, the number of control points used was insufficient to estimate a projective transform, producing estimates that were in fact spurious, and the optimization procedure took more iterations and ended at a cost function value that was higher.

Table 3.1. Seven control-points estimate errors.

	Estimated	**Truth**	**Error**
Easting, m	699660	699654	-6
Northing, m	3956166	3956116	-50
Up, m	186	161	-25
Roll, °	-10	5	15
Pitch, °	33	45	12
Yaw, °	-96	-98	-2

Table 3.2. Two control-points estimate errors.

	Estimated	**Truth**	**Error**
Easting, m	699581	699761	180
Northing, m	3956511	3956998	487
Up, m	150	235	84
Roll, °	16	0	-16
Pitch, °	65	51	-14
Yaw, °	-114	-99	16

3.4 Filtering Image-Matching Algorithm Output with a Kalman Filter

In the previous section, dry runs on sample trajectories reveal that tracks can produce adequate estimates, but appear jumpy and lossy. In a sense, estimating the position and attitude of a UAV by comparing an observed scene image and cross-checking it against a geo-referenced reference image is taking a physical measurement of the location and attitude of a UAV in the world space against its operating environment. As physical measurement processes are expected to have some uncertainty, the raw Image Matching algorithm output are considered to be that raw measurement.

A Kalman filter was then used to post-process the raw IMMAT output. In order to use the Kalman filter, a simple kinematic model is used to describe the UAV's motion. Let **X** represent the position (Easting, Northing, Up) and attitude (ϕ Roll, θ Pitch and ψ Yaw) of the UAV, and that the motion of the UAV can be modelled as in **Equation 3.6**, where **V** is the respective rates of change (in other words, velocity) to be estimated.

This simple kinematic model is sufficient for the trials that were conducted. The trials were all conducted in a race-course fashion with straight legs of constant velocity (test flights

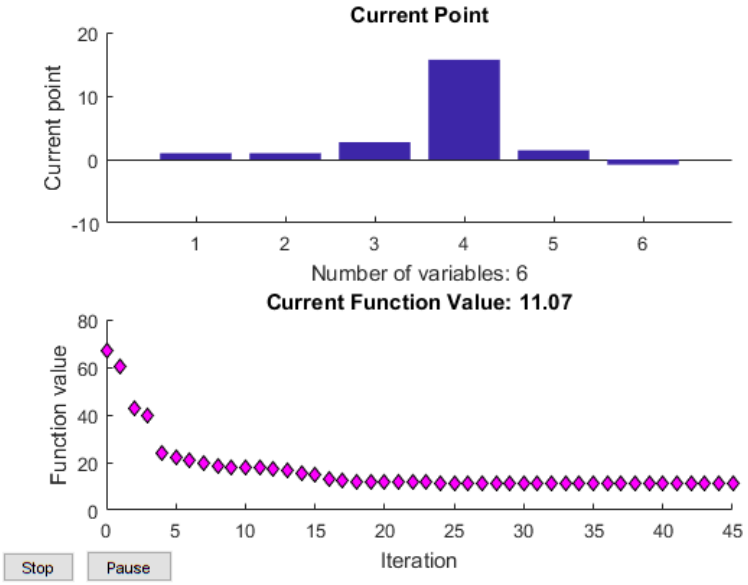

Figure 3.16. Rate of convergence for attitude and pose using seven control-points.

will be described in detail in the next chapter).

$$\mathbf{X} = \begin{pmatrix} E \\ N \\ U \\ \phi \\ \theta \\ \psi \end{pmatrix} \tag{3.6}$$

$$\mathbf{X}_k = \mathbf{X}_{k-1} + \mathbf{V}_{k-1}\Delta t$$

$$\mathbf{V}_k = \mathbf{V}_{k-1}$$

3.5 General Observations

Having walked through the steps of the entire IMMAT procedure, it is possible to summarize a number of factors that can impact the performance of the IMMAT algorithm.

First, drops in the IMMAT algorithm can be caused by various factors, such as insufficient

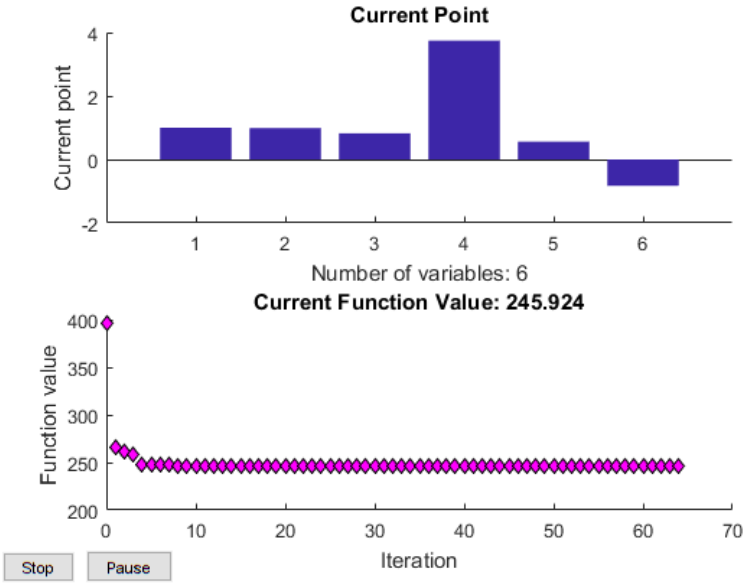

Figure 3.17. Rate of convergence for attitude and pose using two control-points.

matches between the reference frame and the camera image. That could be due to blurriness in the camera images caused by vibration or motion, or because the camera was not looking at the same spot while the aircraft was moving and may have disturbances due to wind or pilot maneuvers. Poor matching can also occur when a reference image is insufficiently rich in extractable features, which can result in low inlier count after rough correspondence matching and then final outlier exclusion through the MSAC algorithm. Finally, the remaining points might be insufficient to estimate a projective transform (which mathematically requires a minimum of four corresponding points).

In order to improve the current performance of the IMMAT algorithm, another modeling method for the re-projection might be used. (For example affine transformation - which although less representative of a perspective view of the terrain, it does require fewer control-point pairs to estimate a transform; this trade-off between accuracy versus generating more estimates offer an avenue for further studies). Also, the RANSAC algorithm culls numerous potential control point pairs in the process of estimating a projective transform. The effect of relaxing the tolerances in the RANSAC algorithm can also be further studied, reducing the accuracy in exchange for generating more estimates that might be useful during the Kalman filtering phase.

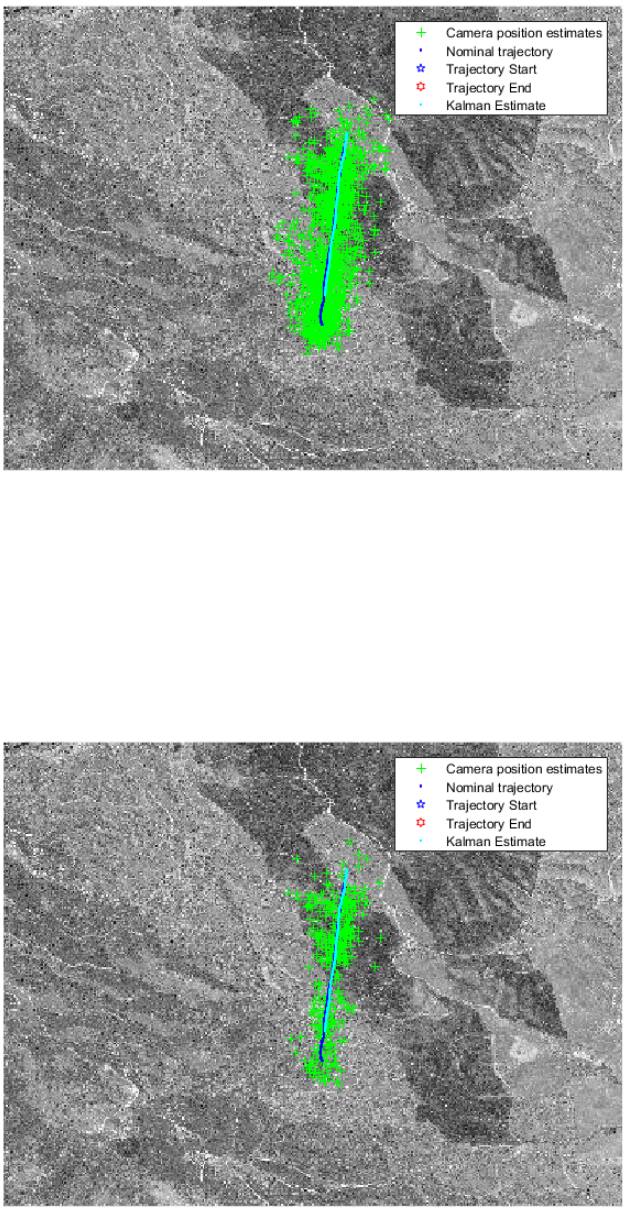

Figure 3.18. Before and after culling points that were generated with insufficient control-point pairs between reference image and camera image.

CHAPTER 4:
Test and Evaluation Setup and Procedures

In the previous chapter, the workings of the image-matching concept were described in detail. This chapter describes the data sets used and the tests that were conducted to evaluate the performance of the IMMAT algorithm used.

4.1 Test Equipment and Data Collection Procedures

This study involved two aerial platforms an unmanned Tier-2 Arcturus T-20 aerial vehicle and a manned Cessna-206, both equipped with the TASE 200 sensor.

According to the manufacturer, the TASE200 sensor is intended to be a compact, lightweight, low cost daylight and infrared camera sensor system. The sensor comes with onboard GPS/INS that allows the system to capture and record ground truth information while in flight. The pertinent specifications of the system are as follows:

- Horizontal Field-of-View: 10.5° for King City recording and 35.26° for Camp Roberts recording;
- Image resolution: 640×480 pixels; however, after interpreting the TASE data, the resolution was found to be 696×464 pixels;
- Embedded GPS/INS sensors
- Camera records at 30 Hz

4.2 Actual Flight Data Collection

Two sets of data were collected: one over King City and another at Camp Roberts. The UAV collected data with the following characteristics:

- cruise speed of UAV
- distance travelled between snapshots
- estimated maximum roll, pitch, yaw and heading changes between each snapshot

The flights were conducted at different altitudes and aircraft attitudes in two different areas,

namely, (1) west of Paso Robles, California, and (2) west of King City, California. Detailed descriptions of the two areas follows:

The first area is within the restricted airspace R-R2504 west of Paso Robles, California. That area has a varying undulating terrain (sample images of the terrain in Camp Roberts is given in **Figure 4.1**), with an elevation of about $300m$. The TASE video stream data were collected at various altitudes as shown in **Figure 4.2**.

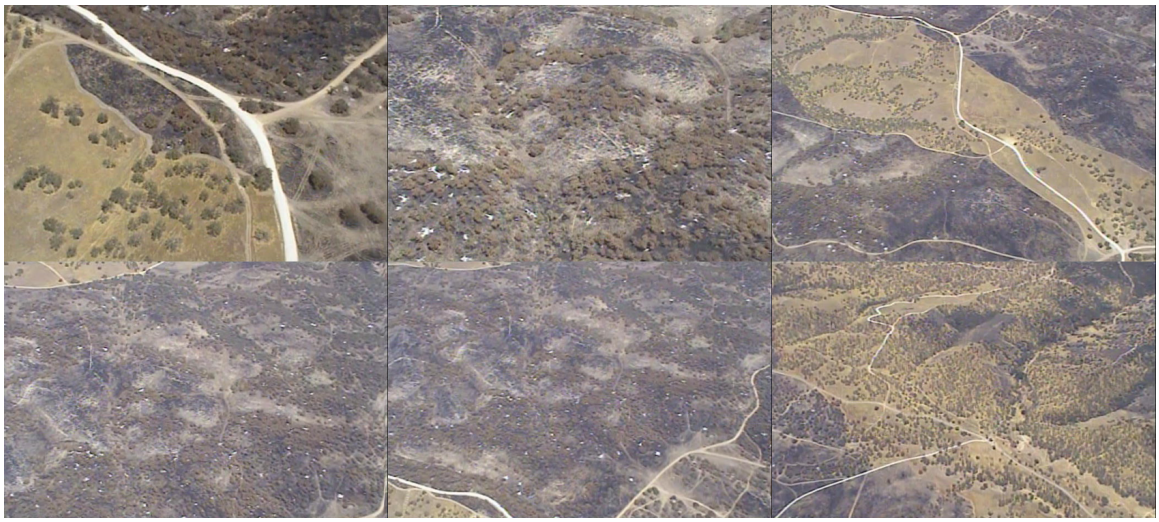

Figure 4.1. Sample images of Camp Roberts' terrain.

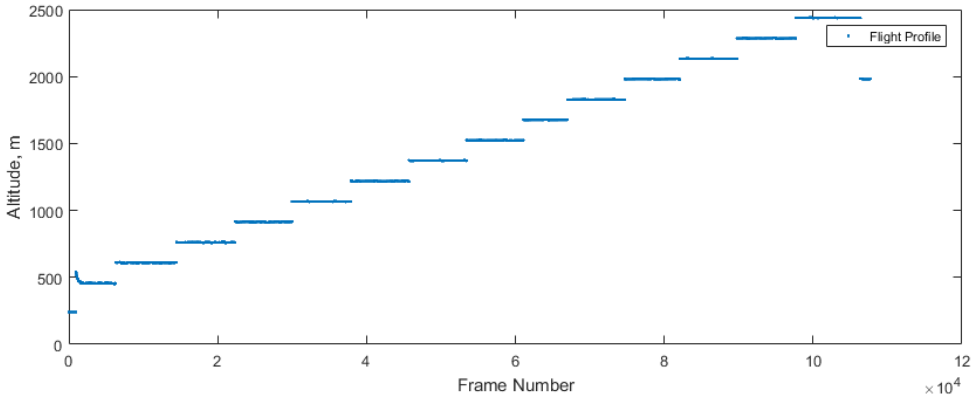

Figure 4.2. Altitude profile versus camera frame number.

At each altitude, the UAV conducted a turn-straight-turn-straight-turn flight profile according to the following (and as shown in **Figure 4.3**):

- one minute straight flight east recording in Electro Optics (EO)
- one minute left turn
- one minute straight flight west recording with EO
- one minute left turn
- one minute straight flight east recording in Infrared (IR)
- one minute left turn
- one minute straight flight west recording in IR
- approximately one minute descent to the next lower altitude

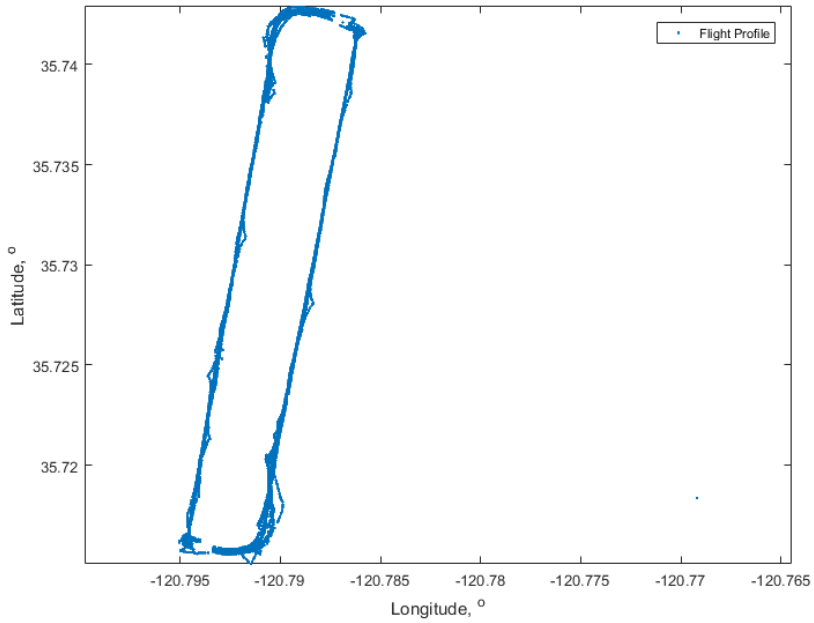

Figure 4.3. Full flight profile for Camp Roberts data collection.

The second area is to the west of King City, California, with a relatively flat terrain, of elevation 100m (see **Figure 4.4**). The area of interest is between Greenfield, California, and King City, California (airport identifier code is KKIC), closer to Greenfield.

The TASE video stream data were collected at altitudes 2000, 4000, 6000 and 8000 feet mean sea level (MSL). At each altitude, the UAV conducts a turn-straight-turn-straight-turn flight profile according to the following (see **Figure 4.5** for an aerial view, and **Figure 4.6** for the three-dimensional view):

Figure 4.4. Flight site at west of King City, California.

- one minute straight flight east recording in EO
- one minute left turn
- one minute straight flight west recording with EO
- one minute left turn
- one minute straight flight east recording in IR
- one minute left turn
- one minute straight flight west recording in IR
- approximately one minute descent to the next lower altitude

The area in King City is nearly flat, with few elevation changes. The terrain is gridded by farmlands except towards the edges that rise up on the Salinas valley. Sample images of the terrain in King City is shown in **Figure 4.7**.

4.3 Preliminary Steps

Data analysis requires the data be separated into different segments of largely similar headings for comparisons. This thesis categorized the data into different UAV headings at different altitudes. This categorization allowed us to evaluate the performance of the algorithm for the same heading direction of the UAV at different altitudes, and likewise, at the same altitude, but flying in different heading directions.

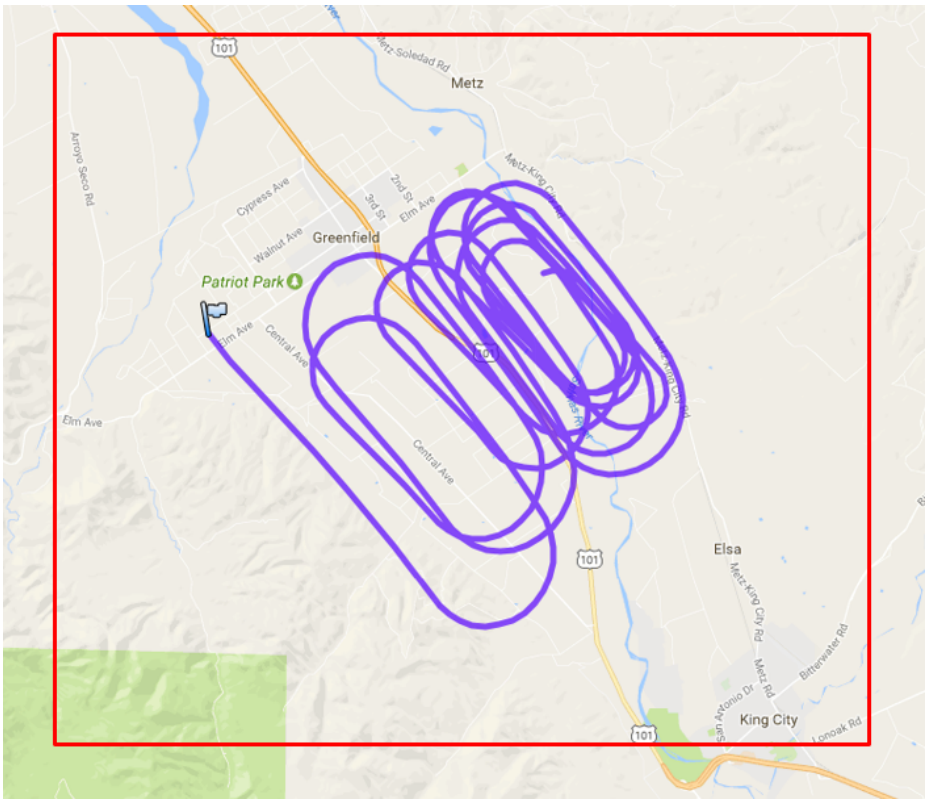

Figure 4.5. Flight profile at west of King City, California.

4.3.1 Data Segmentation

To chunk the data, we used a chunking algorithm to identify segments of flights with largely similar headings (as outlined in the following discussion). Essentially, an initial estimated heading is used to find data points that are within a certain band. These data points are then used to work out the mean (μ) and standard deviation (σ) of the actual raw track. Data points within (2σ) of μ are then designated to be a track.

```
function [filteredData, avg] = filterTracks(data, estimate, band)
    coarseSifting = abs(data - estimate) < band;
    avg = mean(data(coarseSifting));
    TwiceStdDev = 2 * std(data(coarseSifting));
    filteredData = (abs(data - avg) < TwiceStdDev);
end
```

The algorithm operates on the data after taking an initial estimate and a tolerance band

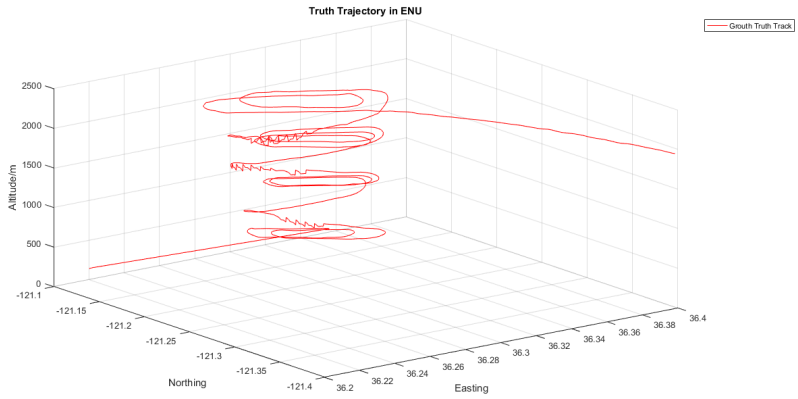

Figure 4.6. 3D flight profile at west of King City, California.

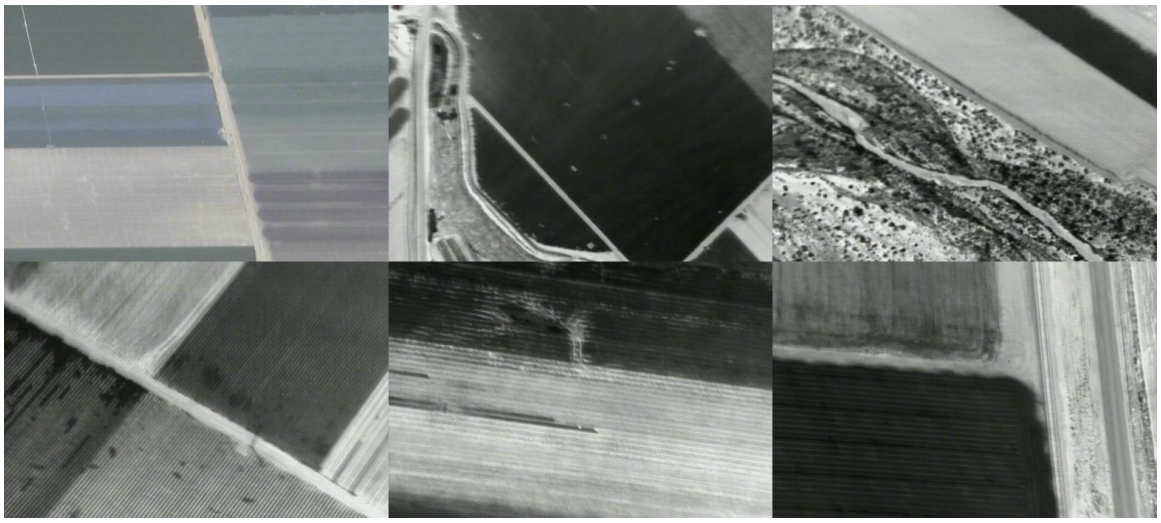

Figure 4.7. Sample images of the terrain over the area between Greenfield, California and King City, California.

coarsely sift out trajectories that might meet the initial estimates. The average and standard deviation of the trajectories that falls within that band are then computed and used to filter data out for those that are within two standard deviations away from the mean heading. An example of the filtered King City data output is provided in **Figure 4.8**, and the alternate view in latitude and longitude plot view is provided in **Figure 4.9**.

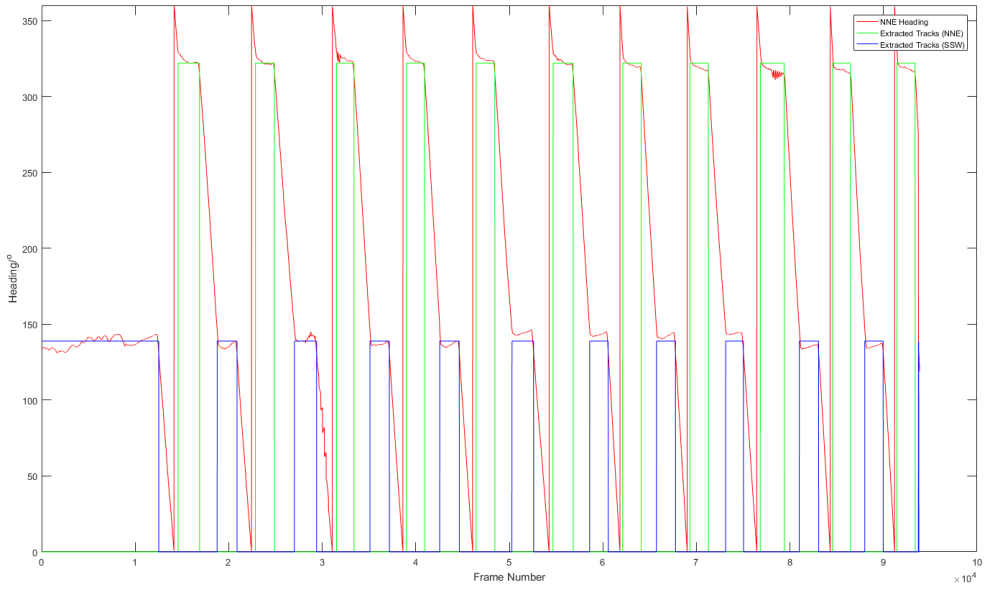

Figure 4.8. King City track segments.

4.3.2 Treating Known Biases in Data

As the equipment was not configured properly during the start of the flights at both Camp Roberts and King City, some ground-truth information logged by the TASE imagers contained biased information.

For the case of Camp Roberts data, the pan of the TASE camera was incorrectly initialized at -90 deg, causing the TASE imager to record the target position off to the left wing of the UAV. In the case of the King City flight, the camera was mounted on the side strut of the UAV, biasing the roll, pitch and yaw values. Data pre-processing were done to remove some of these biases. An example where the viewing target was corrected for is illustrated in **Figure 4.10**.

4.4 Measures of Performances

Measures of Performances (MOPs) need to be developed to quantify and characterize the performance of the image matching approach. The following MOPs were used in the experiments:

Errors in Position and Attitude Estimates. As the image matching approach is for esti-

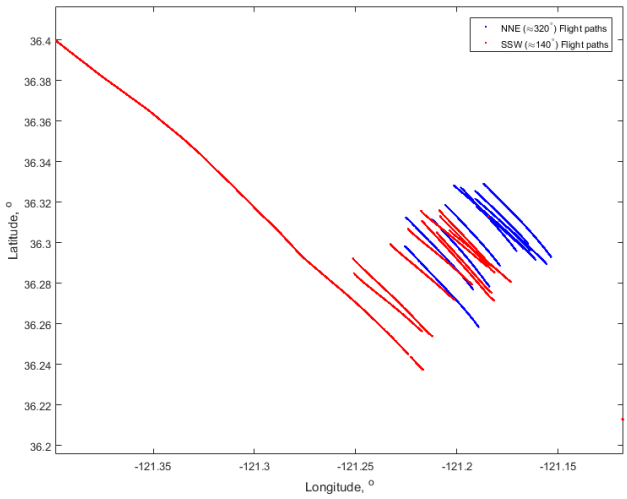

Figure 4.9. King City track segments in Lat-Lon view.

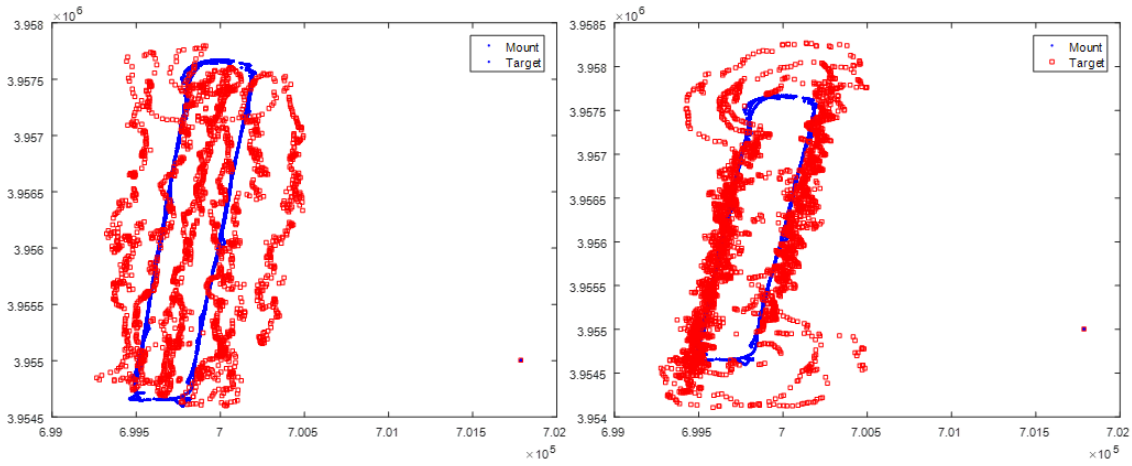

Figure 4.10. Correcting viewing target for data collected at Camp Roberts.

mating the attitude and location of the UAV, the most obvious MOPs are the errors associated with the X, Y, Z position as well as the roll, pitch and yaw of the platform. The performance over various altitudes and over different terrains were studied.

Image-matching Drop Rate. As the algorithm might not find matches all the time due to the real tracks deviating from the nominal trajectory and due to the lack of major image features within either the reference frames or the seeker images, measuring the image-matching drop rate is useful to quantify the stability of the algorithm. Drops were counted when the number of times the IMMAT algorithm used less than five

control-point pairs between the reference image and the camera image.

4.5 Meters per Pixel Resolution

In order to establish whether there is any correspondence between performance of the algorithm and the amount of information that a pixel within the reference frame might cover, it is necessary to work out approximately how many meters a pixel of the UAV camera spans on the physical ground.

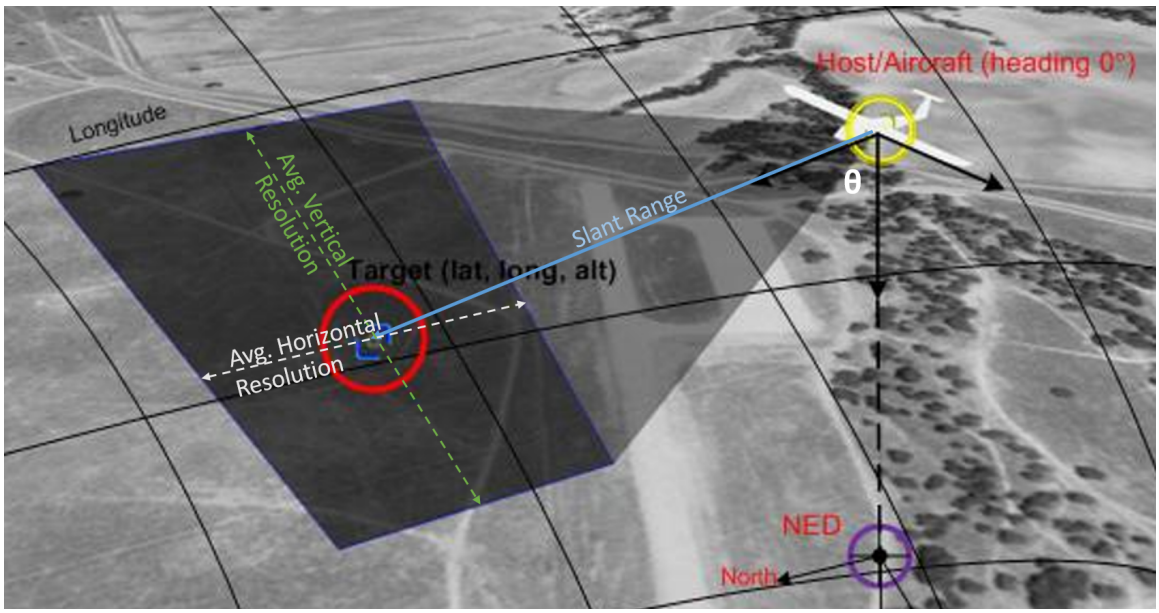

Figure 4.11. Horizontal and vertical ground resolutions.

Referencing **Figure 4.11** it is possible to compute the average distance per meter of coverage of the sensor on the ground as follows:

$$\cos(\theta) = \frac{H}{SlantRange} \tag{4.1}$$

$$\tan(\frac{1}{2}\theta_{HFoV}) = \frac{\frac{L}{2}}{SlantRange}$$

$$L = \frac{2H\tan(\frac{\theta_{HFoV}}{2})}{\cos(\theta)}$$

$$M = \frac{2H\tan(\frac{\theta_{VFoV}}{2})}{\cos(\theta)} \tag{4.2}$$

$$\tag{4.3}$$

In Equations 4.2 to 4.3, L and M are the average horizontal and vertical distances of the projected center width and height of the camera field-of-view respectively, θ is the angle between the camera's direction of view and the normal to the Earth's surface, and H be the above ground level height. The plotted results for the horizontal and vertical coverages are plotted in **Figure 4.12** and **4.13** for Camp Roberts and King City respectively.

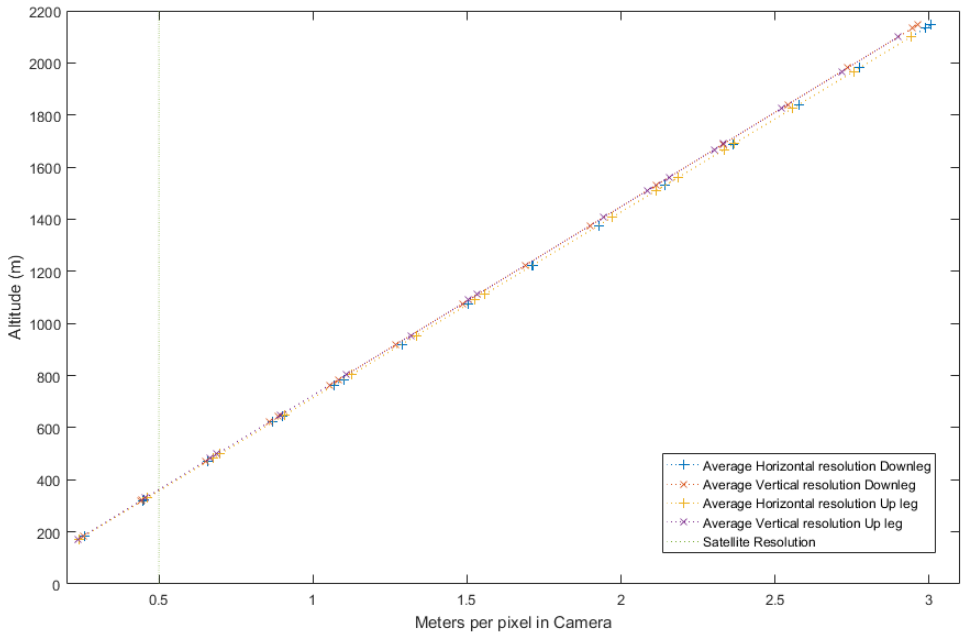

Figure 4.12. Plot of distance per pixel in camera for Camp Roberts flight.

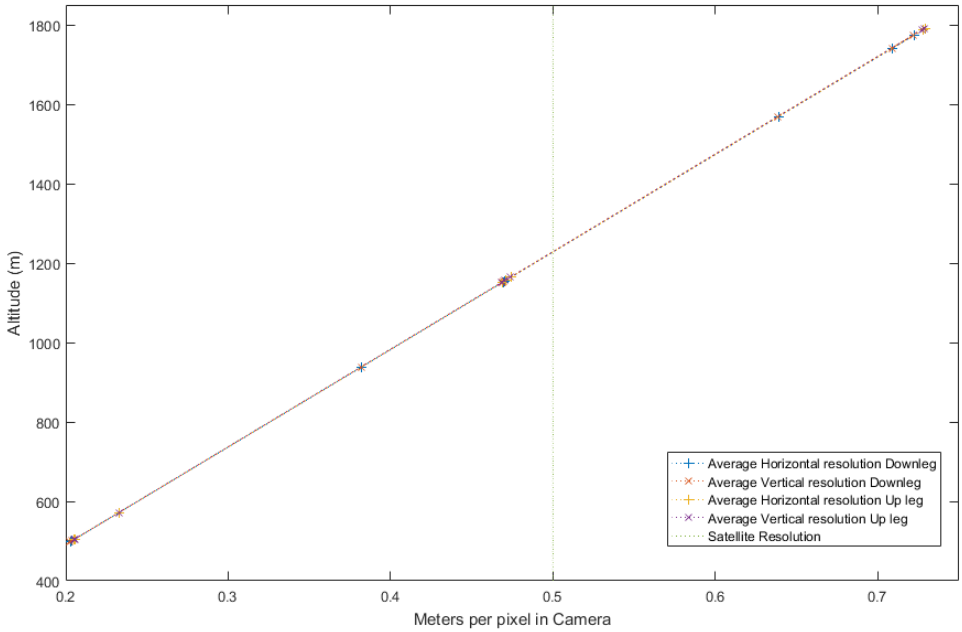

Figure 4.13. Plot of distance per pixel in camera for King City flight.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5:
Data Analysis

In the previous chapter, procedures for analyzing the data collected from test flights were presented. This chapter presents an analysis of the data. After creating nominal trajectories for all tracks that were collected from flights over Camp Roberts and King City, we used the IMMAT algorithm on the video frames to estimate the location and the pose of the UAV. The results generated by the IMMAT algorithm are analyzed in different dimensions in each of the ensuing sections.

5.1 Performance of Algorithm at Different Altitudes

One assumption for the IMMAT algorithm to work was that the projected view of the terrain would be adequate approximation of the camera view for the purposes of image-matching. At lower altitudes, the effects of terrain contouring may be more apparent when the terrain viewed from the UAV's camera. However, the reference images are created from the satellite images which are two-dimensional and can show a different view when re-projected at low altitudes. **Figure 5.1** shows a top view of a terrain which when viewed from a different perspective shows the effect of terrain contours changing the view. This may differ significantly from merely applying a projective transform to the two dimensional top view. Thus, this assumption may be violated should the re-projected view of the planar satellite image differ from the actual perspective view of the physical terrain.

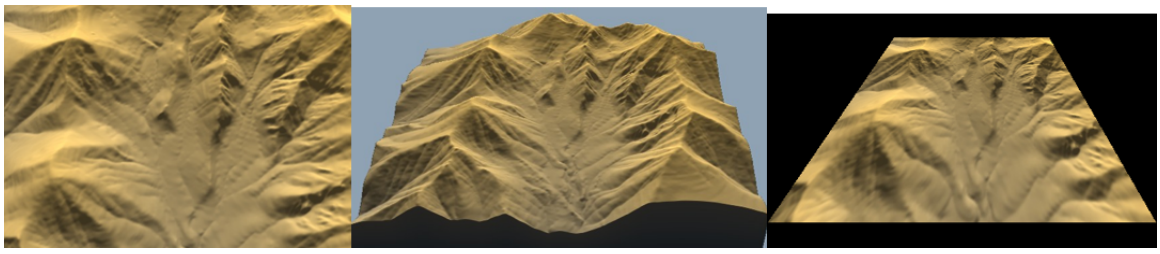

Figure 5.1. Perspective view of the terrain.

The IMMAT output results for up-leg and down-leg flights at various altitudes at different field-of-views used for the generation of reference images are provided in **Figures 5.2, 5.3** and **5.4**, while those captured from King City are found in **Figures 5.5, 5.6** and **5.7**.

Purely by analyzing the mean errors within the Camp Roberts data set, we see the Northing and pitch errors shows a downward bias in errors for both up leg and down leg flights in Easting, Northing and Up positions as altitude increases. Errors in positional estimates are lower at lower altitude.

For Roll, Pitch and Yaw errors, the error experienced in either the up-leg or the down-leg flights appears to increase with altitude. Overall, the errors in Easting, Northing and Up are about $\pm 200m$. Increasing the field-of-views used for the generation of Reference Images does not appear to improve the accuracy of positional and pose estimates.

The variances for estimates in Up increases with altitude, implying that the errors in estimating altitude may be proportional to the altitude of the flight.

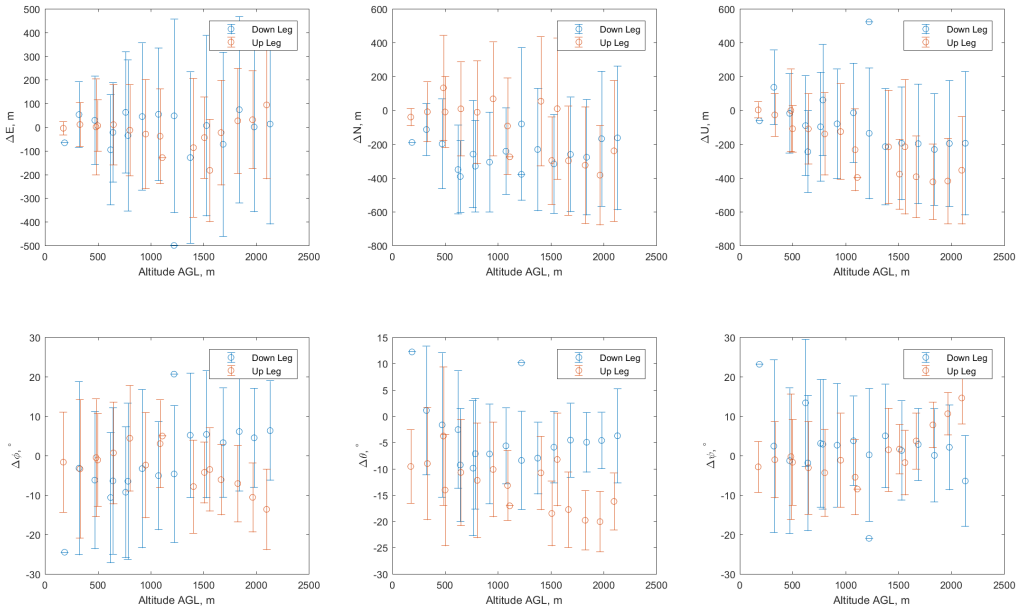

Figure 5.2. Camp Roberts error plots at various altitudes with Reference Frames matching at 1x FOV.

Analyzing the King City results, we find the data shows that the mean errors for Easting and Northing are around $\pm 200m$. Above-ground-level altitude is also in line with the Camp Roberts data at $\pm 50m$.

Figure 5.8 shows the graphical outputs of the IMMAT algorithm at various FOVs for a

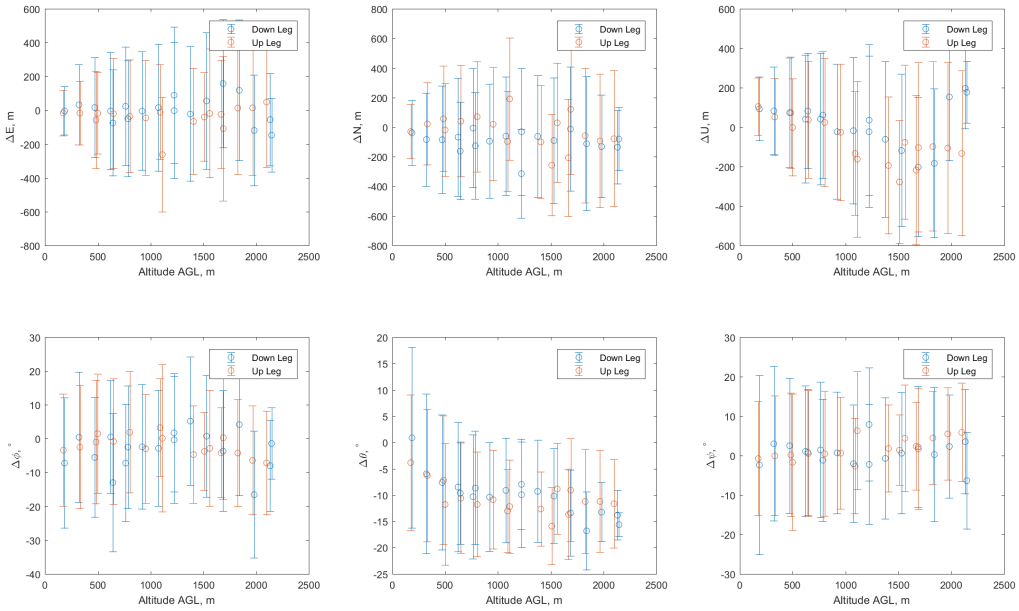

Figure 5.3. Camp Roberts error plots at various altitudes with Reference Frames matching at 2x FOV.

down leg track flying at 600 feet AGL. As the FOV of the Reference Images increases, covering more of the terrain, the overall number of potential IMMAT matches increases, leading to a progressively denser plot of estimated position.

In general, images of the underlying terrain needs to be sufficiently feature-rich for the IMMAT algorithm to work. For the case of trajectories captured in King City, the feature counts in the reference frames were themselves generally low. **Figure 5.9** shows the number of features that were extracted from each of the reference image and each camera image throughout an entire example trajectory (down-leg flight, at 5147 feet AGL). The track displayed visually is found in **Figure 5.10**. To begin with, the number of features fell below 800, averaging about 300 before coarse correspondence matching. The numbers after matching and RANSAC results in no inlier matches for nearly the entire trajectory.

The satellite images used for this study over King City come from about half a year after the actual flight was captured; data sets closer to the actual date of flight were not used as they had cloud coverage that occluded land features, which are required for the IMMAT

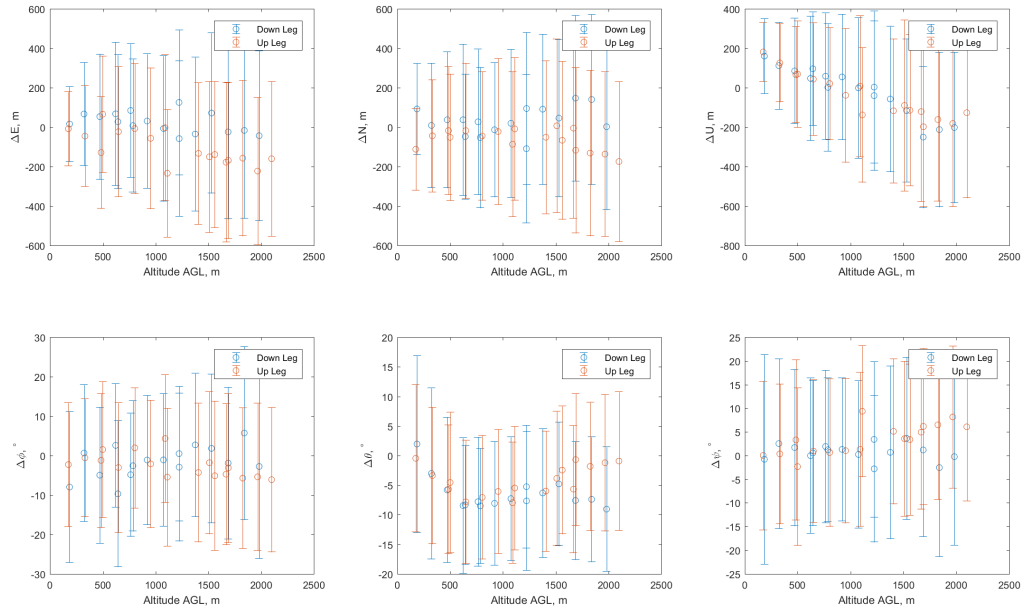

Figure 5.4. Camp Roberts error plots at various altitudes with Reference Frames matching at 3x FOV.

registration algorithm to work.

It was observed that the IMMAT algorithm was latching onto permanent and prominent terrain features such as rivers or hills which does not change too much with time. **Figure 5.10** shows the same trajectory as previously described for the case with low feature counts. The stored reference frame showing a distinctive and prominent landform, in this case a river.

5.2 Effect of Reference Image Field-of-View

Intuitively, we surmise the bigger the field-of-view of the reference image, the higher the likelihood of it containing the potential views of the in-flight camera. The photo montage in **Figure 5.11** shows the reference frames (represented by the green patch) generated at 1x, 2x and 3x FOV respectively. The black triangle and magenta track shows the view point of the in-flight camera. For a small FOV, it was usually difficult for the reference frame to contain the view direction of the actual flight viewing position. At larger FOVs,

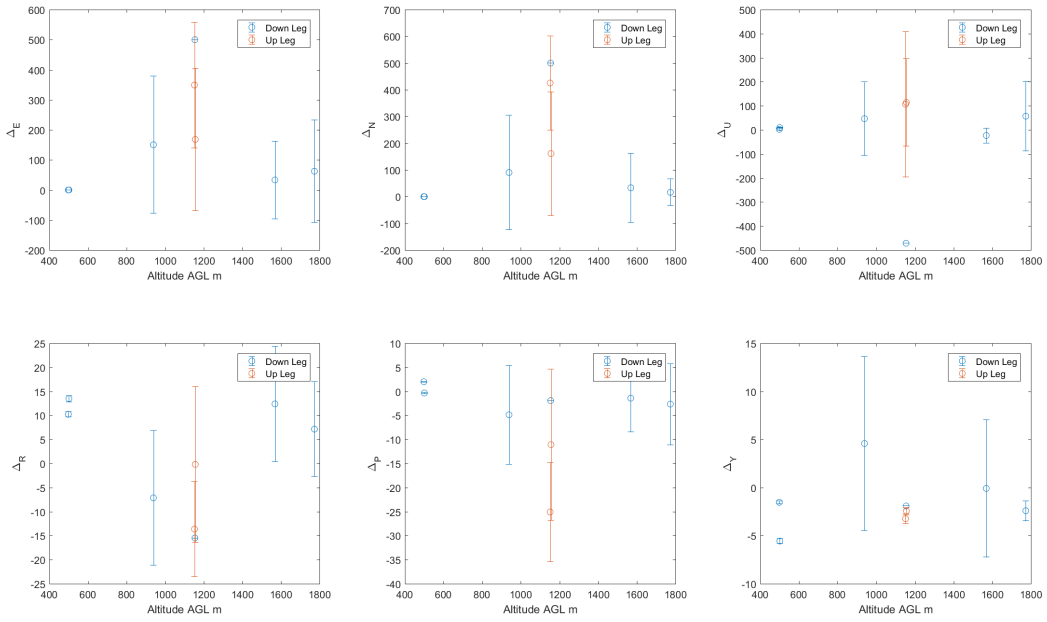

Figure 5.5. King City error plots at various altitudes with Reference Frames matching at 1x FOV.

the view points are usually contained inside the green patch (which will be projected into a rectangular reference frame).

In general, increasing the field-of-view for the reference image generation directly lead to an increase in the numbers of features that can be extracted from the RIL images. **Figures 5.12, 5.13** and **5.14** shows the that when the field-of-view used to generate the reference frames increases, the number of features found in the reference frames increased from an average of around 1000 to around 10000 features. The number of inlier matches also increased on average with the increase in the Field-of-View size. While it would be useful also to study the effect of increasing the field-of-view of the actual camera view, live data is not available at the time of this study.

5.3 Drop-Rates of the Image Matching Algorithm

In the previous section, data suggests that within the altitudes of the data set (lower than about $2km$) the accuracy of the IMMAT algorithm does not really vary with the different

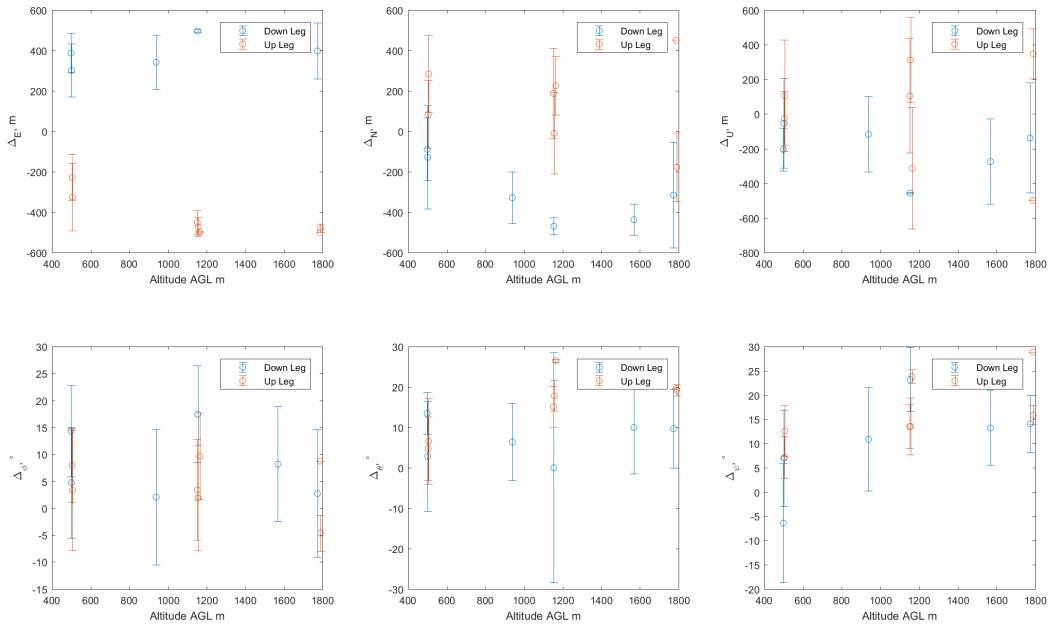

Figure 5.6. King City error plots at various altitudes with Reference Frames matching at 2x FOV.

altitudes.

Another aspect that was important for the IMMAT algorithm to be operational is the drop-rate. As discussed in the previous chapter, a drop is assessed when the number of control-points used to estimate the location and pose of the UAV is less than five. An example of the outcome of drop rates at various altitudes for a Camp Roberts flight is found in **Figure 5.15**. On the whole, the drop rates appear to increase with altitude. Drop rates are generally high; on average 80 percent of the points do not have sufficient control points for the IMMAT algorithm to work.

5.4 Distribution of Image Matching Predictions

Analyzing the graphical output of the IMMAT algorithm we find significant information about how the data are distributed. The outputs of an unconstrained search are briefly discussed in the ensuing paragraphs, and then a more detailed discussion for the constrained search results is provided.

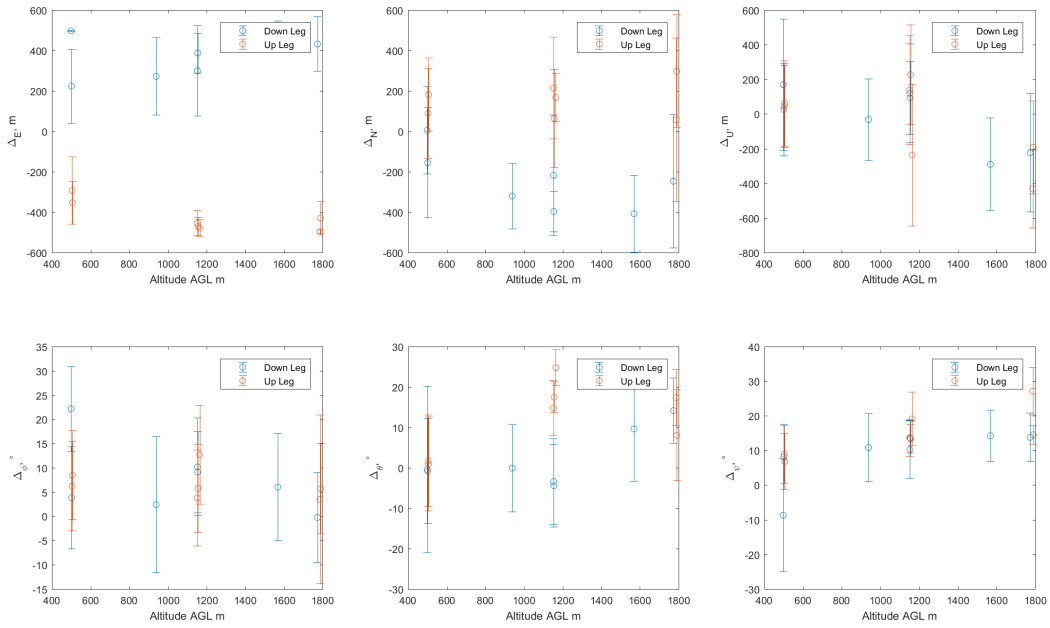

Figure 5.7. King City error plots at various altitudes with Reference Frames matching at 3x FOV.

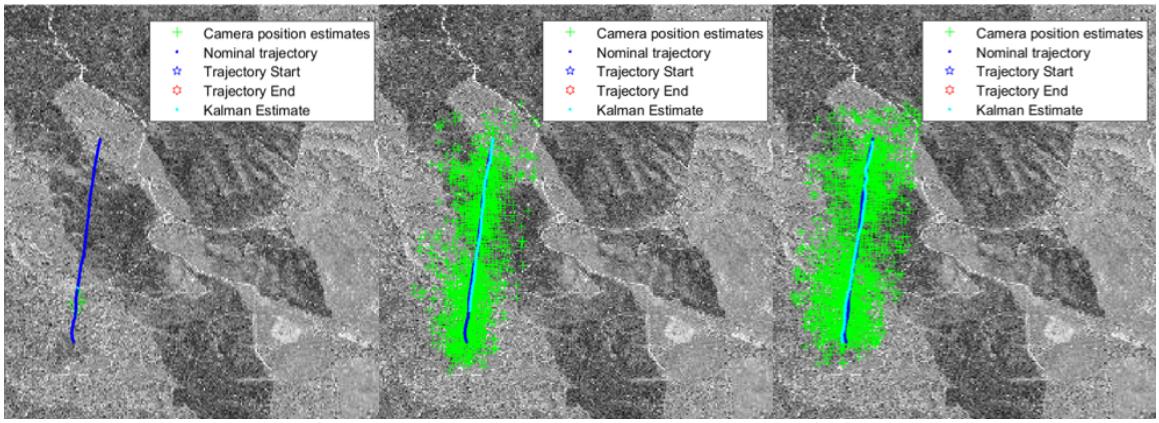

Figure 5.8. Down leg track output for three different FOV sizes for Reference Frames.

The predictions for the position of the aerial camera were observed to spread about the nominal trajectory for all tracks that were analyzed (see **Figure 5.16**) when using an unconstrained search. The predictions are sparse in comparison to the constrained search (discussed next), and when an estimate is offered, it is noisy and jumpy. This is evidenced

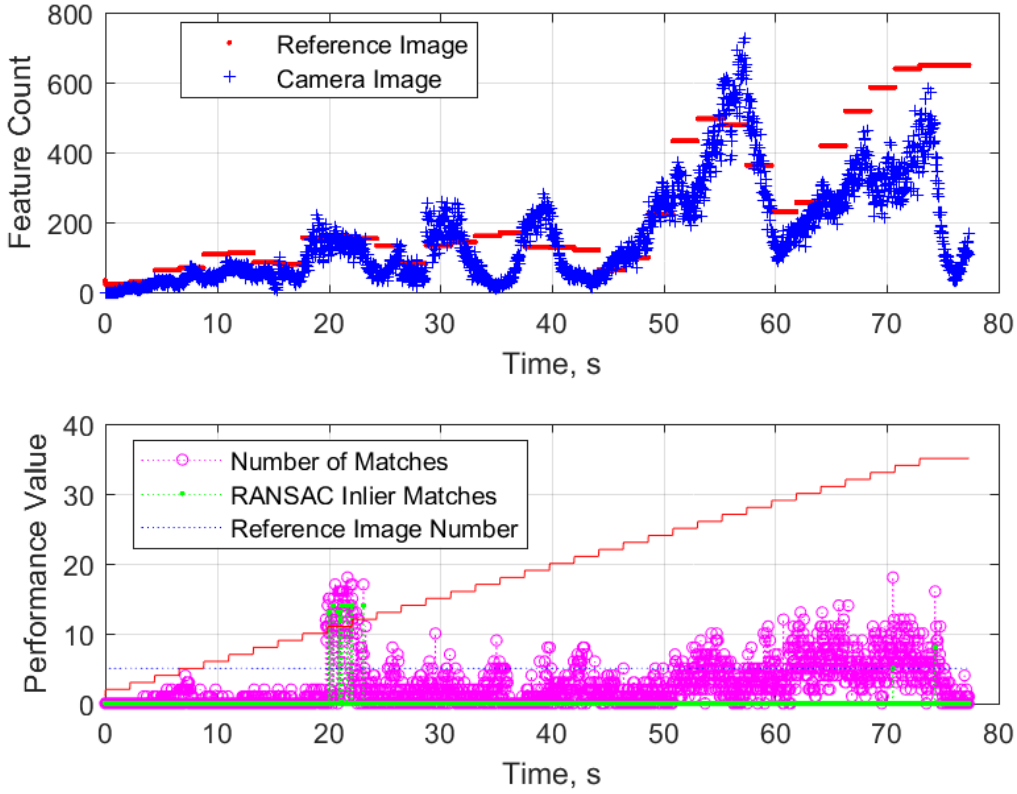

Figure 5.9. Generally low feature counts for King City trajectories.

by the Kalman filtered track providing a predicted trajectory that was inaccurate due to the unstable feeds coming out from the unconstrained search.

Using a constrained search we observed a significant improvement in terms of producing estimates (see **Figure 5.17** for a typical output) as well as improving the accuracy of the estimates. For the sample track that was illustrated, when the IMMAT algorithm is able to find an estimate close to the nominal trajectory as the optimal, the algorithm was able to provide a very accurate estimate for position. However, should the constrained search not be able to find a solution it eventually reached the boundary of the `fmincon` search. As the Kalman filter track is unable to distinguish between a good or a bad IMMAT estimate, the Kalman filtered track also uses those predicted points along the boundary of the constrained search but moves towards it as there were a lot more points there. This offers an opportunity to filter those outliers easily.

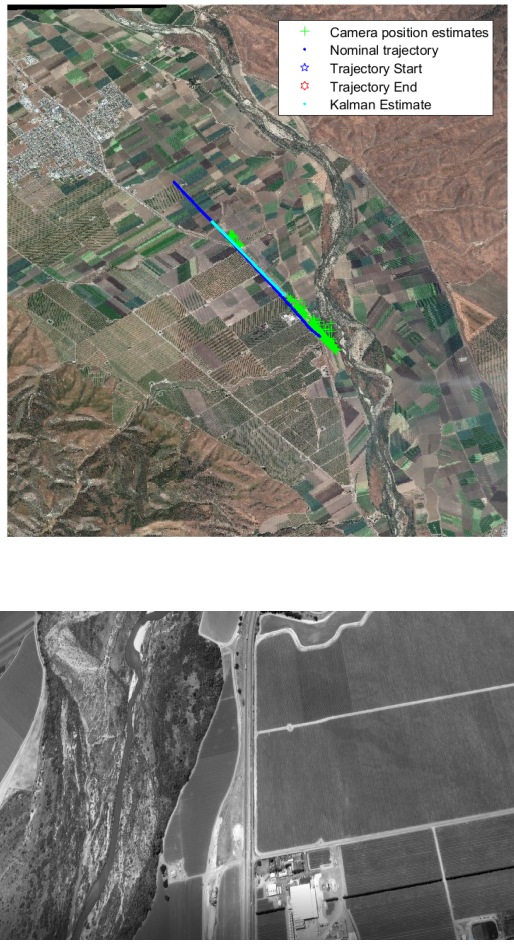

Figure 5.10. Salinas River a prominent and distinctive landform.

Figure 5.11. Different field-of-views used in Reference Images generation.

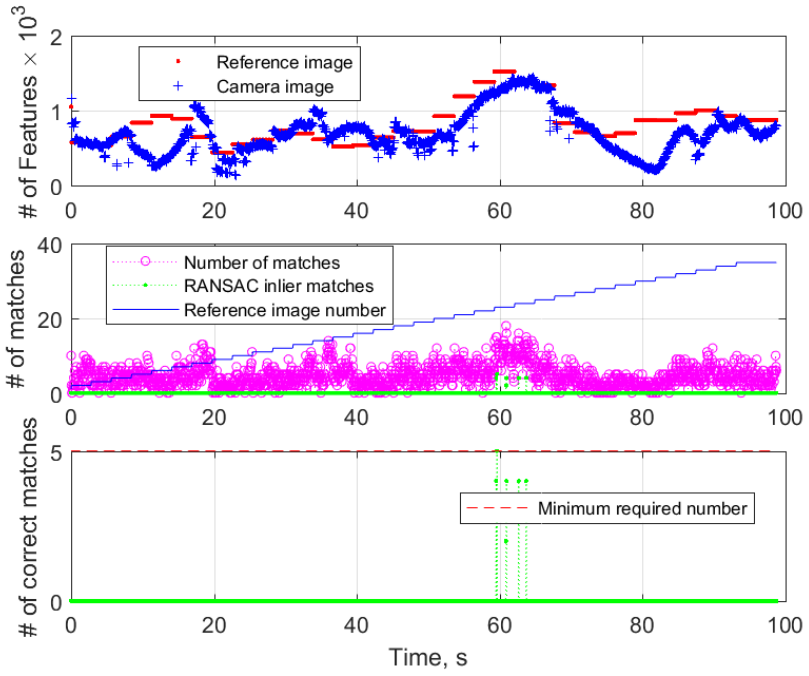

Figure 5.12. 1x FOV for Reference Images generation.

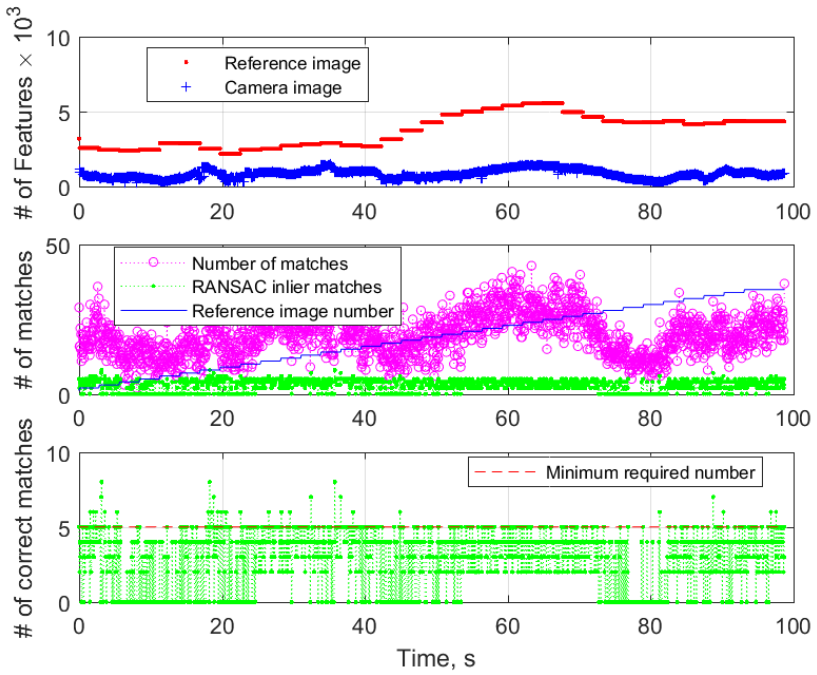

Figure 5.13. 2x FOV for Reference Images generation.

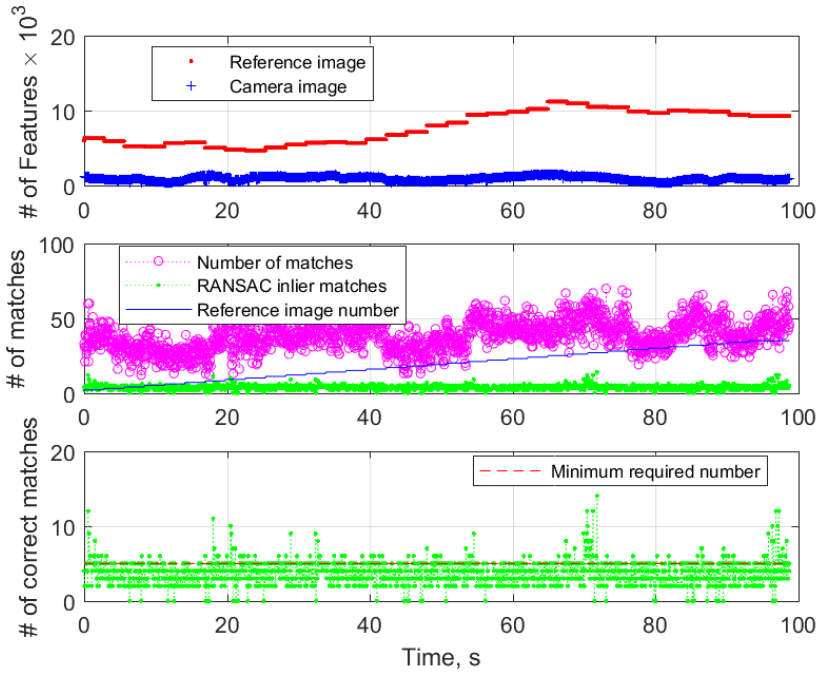

Figure 5.14. 3x FOV for Reference Images generation.

5.4.1 Effect of Creating Reference Frames with Larger Simulated Field of View

The underlying assumption is that with a larger reference frame, the likelihood of capturing the real camera scene within the frame will increase Secondly, due to the higher likelihood of full overlaps, the number of inlier matches will also increase, increasing the probability of getting an IMMAT match. The analysis results bear out these assumption;, with a sample output for a Camp Roberts trajectory shown in **Figure 5.8**, the number of position estimates increases when a larger field of view is used to generate the reference image frames.

5.4.2 Performance of Image Matching Algorithm for Different Terrains

The King City and Camp Roberts data sets cover two different terrain profiles the former being a flat terrain covered with farms and the latter being a hilly undulating terrain.

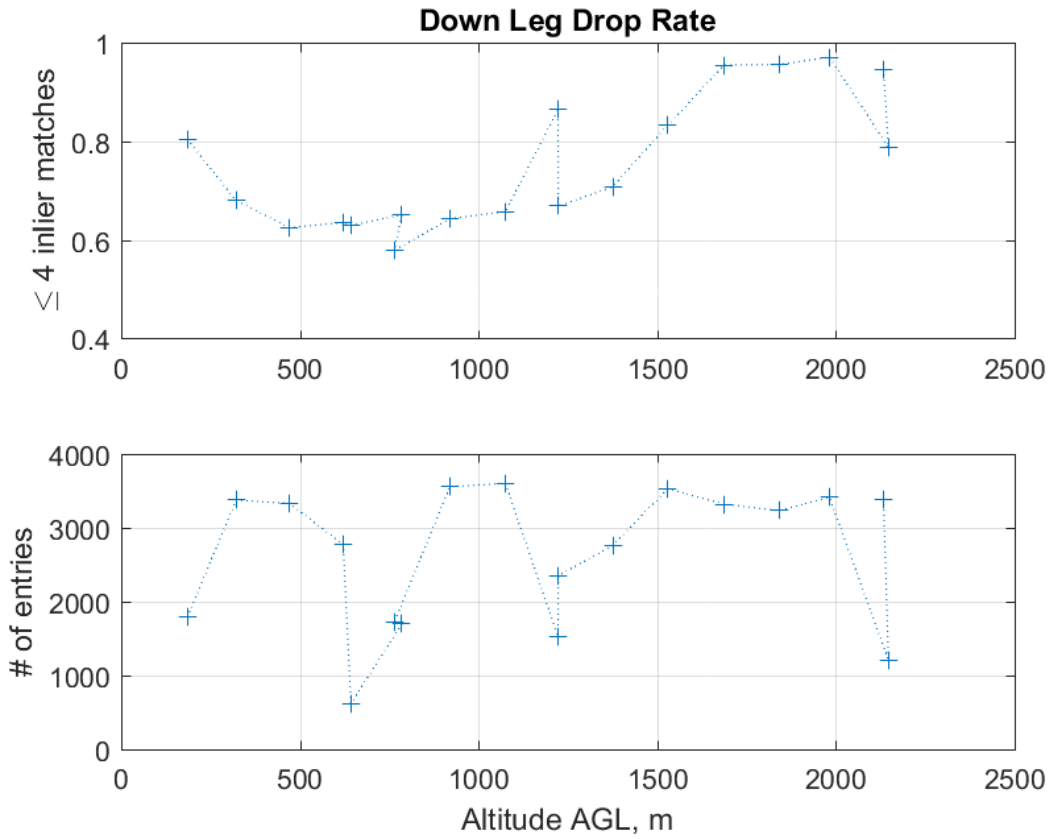

Figure 5.15. Sample drop rates of IMMAT algorithm for Camp Roberts flights at various altitudes.

5.4.3 On Flat Terrain

For terrain that exhibit repetitive patterns (in the case of King City - fields in one area look largely similar to the gridded field structure in another area), or when the terrain lacks any distinctive features that might allow it to be distinguished from another area, the IMMAT algorithm will fail to produce a match, contributing towards the drops.

At low altitudes and small field-of-view, terrain images that were captured in King City lacked distinguishing features, leading to high drop-rates and ineffective IMMAT matching.

In King City however, the number of spurious matches are significantly lower when compared with Camp Roberts.

Terrain images that were captured at low altitudes and with a small field-of-view in King

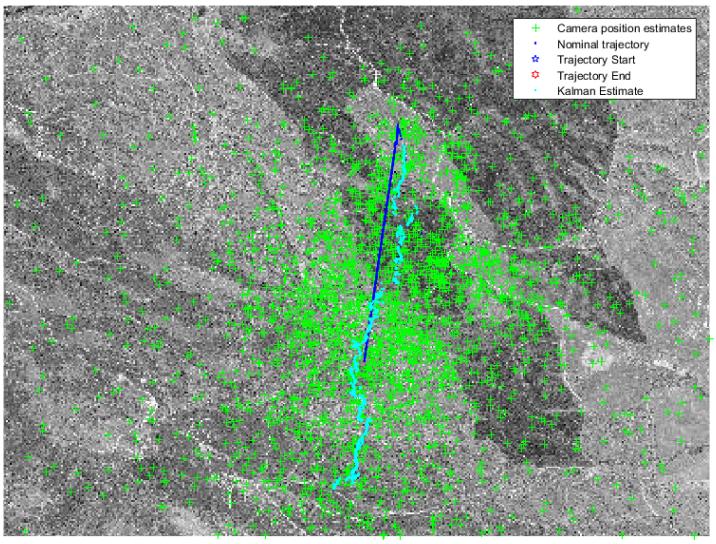

Figure 5.16. Typical appearance of an IMMAT output by unconstrained search.

City lacked distinguishing features, which led to high drop-rates and ineffective IMMAT matching. Some sample scenes where there are insufficient salient features in the images are given in **Figure 5.18**.

In King City, however, the number of spurious matches is significantly lower when compared with Camp Roberts.

5.5 Analyzing Data Generated at Various Altitudes and in Different Flight Directions

The IMMAT output results for up-leg and down-leg flights at various altitudes at different field-of-views used for the generation of the reference images are provided in **Figures 5.2 5.3** and **5.4**, while those captured from King City are found in **Figures 5.5, 5.6** and **5.7**.

Increasing the field-of-view of the reference images led to a greater number of matches between control points in the reference image and the camera image but did not appear to improve the overall accuracy of the positional estimate.

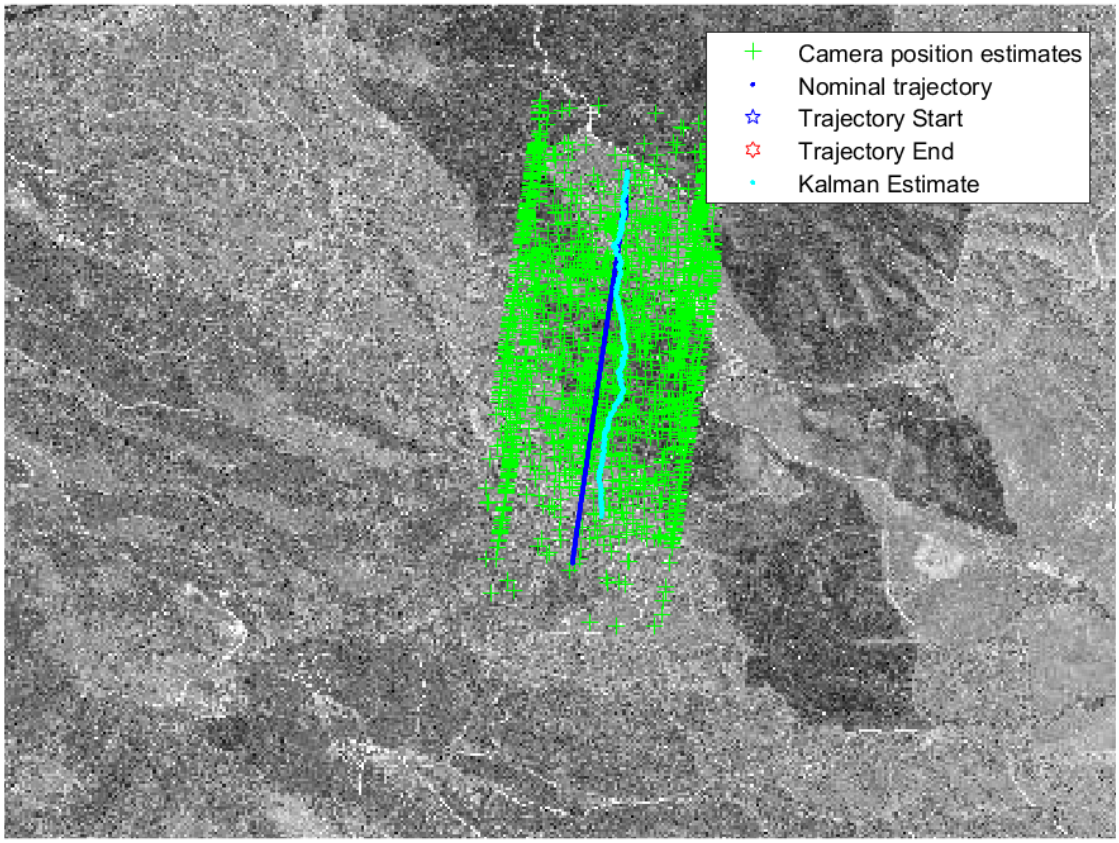

Figure 5.17. Typical appearance of an IMMAT output by constrained search.

While the drop rates within the King City data sets for smaller field-of-view were significant and led to limited analyzable information, the errors in up-leg and down-leg flights showed a clear differentiation between Easting, Northing and pitch estimates. The reason why the

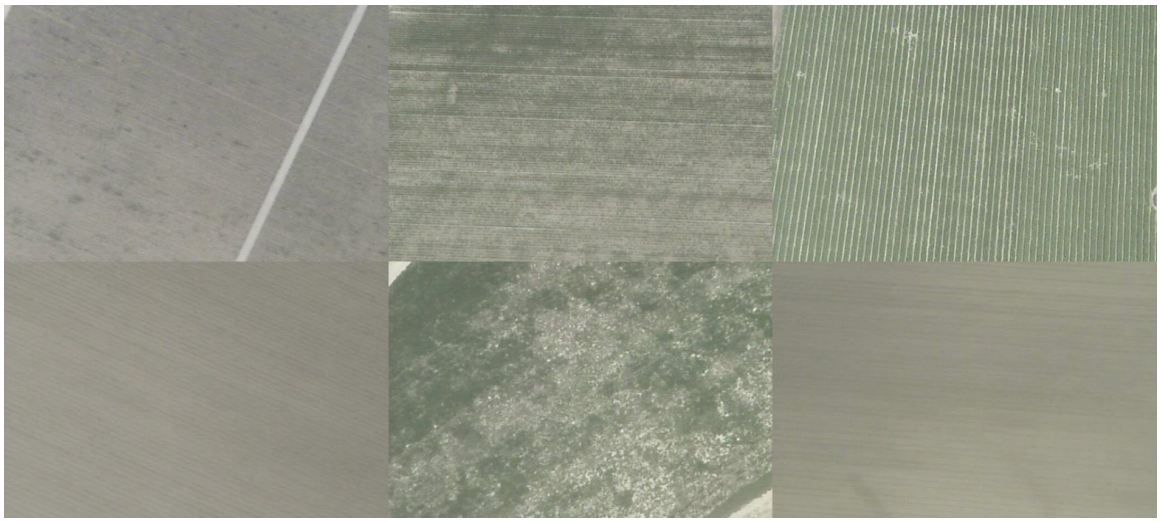

Figure 5.18. Featureless terrain.

estimates separate out the way they do could be due to systematic biases, which require further investigation.

Analyzing the King City results, we observe the data shows that the mean errors for Easting and Northing are around $\pm 200m$. Above-ground-level altitude is also in line with the Camp Roberts data at $\pm 50m$.

The underlying assumption is that with a larger reference frame, the likelihood of capturing the real camera scene within the frame will increase, and secondly, due to the higher likelihood of full overlaps, the number of inlier matches will also increase, increasing the probability of getting an IMMAT match.

Figure 5.8 shows the graphical outputs of the IMMAT algorithm at various FOVs. As the FOV of the Reference Images increases, covering more of the terrain, the overall number of potential IMMAT matches increases, leading to a progressively denser plot of estimated position.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 6:
Conclusions and Future Research

This chapter concludes the thesis report. It begins by summarizing the work done for this thesis, the proceeds to review the key conclusions presented in the previous chapter. This chapter closes with a consideration of the limitations of this research and provides suggestions on further studies.

6.1 Summary of Work Done

This thesis enhanced our understanding of how to deploy image-matching algorithms for guided unmanned activities that may operate in a predetermined area, following a planned trajectory. In such a case, recently captured high-resolution images of the operational environment over which the planned trajectory is expected to fly can be pre-loaded onto the unmanned system. This information can be used as an alternative navigational aid when other on-board navigational equipment fails or cannot be used. One specific application for which this capability will be useful is in autonomous military operations within a GPS-degraded or a GPS-denied environment.

This thesis was motivated by the possibility of leveraging camera sensors that are commonly available onboard UAVs to provide an alternative source of positional estimates. The purpose of pursing this approach was to develop an alternative should other sources of location feeds fail to provide updates. Conditions warranting such an alternative include when the GPS fails to work due to area denial, or when the IMU drifts too much due to various aerial maneuvers or because the IMU has not received current positional updates. The approach taken for this thesis work relies on the preliminary study conducted in 2016 by [3], in which they described an idea to match a camera's view with a geo-referenced library of reference images. This thesis extended the work done previously by conducting a functional analysis on the image matching navigation problem following Systems Engineering best practices [7], to better frame the problem. A list of MOPs was also established to better characterize the behavior, performance and applicability of the IMMAT algorithm.

Having better framed the task, we proceeded to test the IMMAT concept further by conduct-

ing experiments on the core IMMAT algorithm based on flights held in different locations and at different altitudes. To evaluate the behaviour of the image matching real flight data captured in King City and Camp Roberts, California, were used for data analysis. This data came tagged with, most importantly, the ground truth GPS location of the platform as well as the attitude of the platform at a specific moment in flight.

Five major observations from the conducted evaluations are as follow:

1. The IMMAT approach relies on the feature-richness of both satellite and onboard camera images. To this end a typical satellite image provides a resolution of 0.5 square meters per pixel regardless of the size of the ground footprint. Resolution of the on-board camera depends on the field of view (zoom setting), altitude and attitude. The best resolution is achieved in a level straight flight at low altitudes with a maximum zoom in. However, such a setting results in a very narrow field of view (significant reduction in the number of features that can be used to match those of the satellite image). Specifically, with the TASE-200 sensor used in this research and a FoV of 35 degrees (Camp Roberts flights), a resolution of $0.5m^2$ per pixel can only be achieved when flying below 400m AGL. Likewise for King City flights, where the videos were taken at FoV of 10 degrees, only flights below 1200m can achieve $0.5m^2$ per pixel resolution.
2. The texture of the surface has a major influence. Specifically, flying over the agricultural area (between Greenfield and King City) at low altitudes with a narrow FOV results in no features detected in the onboard camera field of view when crop fields are under the flight path. Some features can be detected only when flying in between the crop fields. One way to mitigate this effect might be increasing the FOV, but that leads to decrease in resolution and possible failure to find the matches between two different resolution images. Still, this approach is worth exploring in the future.
3. Onboard camera stabilization (suppression of vibrations) plays a crucial role as well. In this research two aerial vehicles were used. The same sensor, a TASE-200, had much better stabilization when flying on the UAV at 25m/s compared to that of the manned Cessna-206 flying twice as fast.
4. Varying terrain elevation also influences the accuracy of the IMMAT navigational solution. That includes a requirement to have a detailed terrain elevation map of the intended area of operations.

5. Aircraft attitude plays a major role as well. In this research IMMAT performance was evaluated only for the straight level flight. Future evaluation should consider IMMAT performance while turning / climbing / descending.

This research used a limited set of test data based on a TASE-200 sensor, which is not a high-end device. The sensor had some vibration isolation problems along with incorrect reporting of pan-tilt information (which was discovered within this research effort and reported to a manufacturer) resulted in an unusually high drop rate. This occurred when there were not enough matching points to construct a projective transformation, which is a basis of the IMMAT approach. Nevertheless, this thesis was able to conduct a detailed assessment of the overall performance of the IMMAT algorithm.

The main conclusion is that when all conditions are met (i.e., at least five matching points are found), the IMMAT algorithm can provide an estimate of aerial vehicle position as good as within $50m$ from its true position (this value correlates with the satellite image resolution), and determine its attitude within ± 15 degrees for pitch and roll while finding its yaw angle within just ± 2 degree accuracy.

Some additional observations follows.

- For the same field of view, as the flight profile increases in altitude allowing more of the local terrain to be captured, with a consequential increase in the number of features and the likelihood of matches, the drop rates for the IMMAT algorithm decreases.
- If an IMMAT drop does not occur, then the error associated with IMMAT estimation decreases with the altitude or pixel-per-meter on the ground.
- This thesis relies on a simple two-dimensional projection of satellite imagery into the view of a would-be camera in flight. The lack of elevation data introduces perspective differences that may contribute to the errors in estimation by the IMMAT algorithm. To further quantify the errors due to projection, there are two further experiments that can be conducted. First, real video imagery can be taken at various tilt angles, with the most important being vertically downwards. The downward view matches best with the top-down satellite view and also obviates the need for terrain elevation information for projection purposes. The second is to enhance the projection algorithm by capturing a view from a three-dimensional satellite image textured digital elevation model from the perspective of the camera, and comparing

the estimates with the current approach.

- While the RIL can be created from a large collage of high resolution satellite images prior to flight and then stored onboard the UAV, it can require quite a bit of space to store the frames. For example, a nominal trajectory that requires about 700 reference frames stored in high resolution amounted to 0.5GB; storing only the extracted features and using only those features will require much less space. This presents an opportunity to investigate a method for storing the that can work with the image-matching algorithm efficiently with the image-matching algorithm.
- As the IMMAT algorithm produces an estimate frame-by-frame and only when sufficient matches are found, there will be variations in the estimates generated when they are produced, otherwise there are no estimates. The question is whether feeding the output of the IMMAT algorithm into a Kalman filtering process (1) produces a cleaner output, (2) produces (hopefully) more accurate positional predictions, and lastly (3), to use the previously known positional predictions to feed as an initial positional estimate into the 6-DoF optimization procedure.

6.2 Future Development

There are various opportunities to study areas where the entire image-matching navigation procedure can be optimized. One area of possible further study is to optimize the number of reference frames, answering the question on what would be the minimum number of frames required, below which the performance of the image-matching algorithm degrades. One possible idea is to take advantage of the fact that the code-base today is able to plot the viewpoint of the camera onto the aerial view of the area of operations. Using this information, it is possible to work out how far apart the reference frames can be spread out and still contain the viewpoint of the camera. The algorithm as designed today generates reference frames based on a nominal trajectory that has been divided up into evenly spaced segments, and then generates a projection at those points on the ground, given the UAV's nominal pose.

6.2.1 Creating a Feature Rich Reference Image Library

During the creation of the RIL, there were instances where the reference images selected had few features. These frames were still included in the RIL to keep the algorithm simple,

so that the thesis could proceed and investigate the image matching performance of the 6-DoF UAV pose estimation procedure instead. This is therefore one area of immediate future work where a technique can be developed for selecting reference image frames that have sufficient features, but yet sufficiently spaced out and representative of the nominal trajectory to be covered. In so doing, the drop-rates for the image matching algorithm will be immediately reduced, improving the stability of the image-matching navigation process.

Another area of study is on the skip-rate of the incoming video stream, to answer the question of how many frames in an incoming video stream can be skipped to avoid unnecessary processing, but still allow it to provide accurate estimates on the UAV's position.

6.2.2 Investigating Image Feature Extraction Ability of Various Algorithms for Different Terrain Types

In the previous section, data supports the claim that drop-rates are highly associated with the feature extraction capabilities of the image feature extraction algorithm used, if the scenes between the reference image and the camera view are indeed overlapping.

As the feature extraction algorithm is a component that can be substituted, future work can investigate the use of other feature extraction schemes such as SIFT or BRISK. Such work can investigate which extraction can investigate which extraction methods are appropriate for the various terrain types.

6.2.3 Managing Drops in Image Matching

On the whole, during the batch processing of the data, high IMMAT drop-rates were observed —with some tracks reaching 100%. Continued work to reduce the drop rates needs to be done to improve the robustness and reliability of the current IMMAT algorithm so that it can function as a viable source for navigational updates.

During the image-matching procedure, some scenes may not provide adequate feature pairings for the attitude of the UAV to be estimated. The circumstances under which drops may happen could be due to various reasons (those that are known were previously discussed in **Section 3.3.1**), but more flight data over different types of terrain will be useful to ascertain whether it might be the performance of the feature-extraction algorithm that is

affecting the overall performance, and whether the feature-extraction algorithm is terrain dependent. Knowing this information will be useful, and can be done ahead of time, for tuning the algorithm prior to any unmanned flights. This thesis relies on using the SURF algorithm for feature extraction, so further investigation can be conducted using different feature extraction algorithms for areas where the SURF algorithm gave poor results.

It is assumed that the projected view of the terrain is an adequate approximation of the camera view for the purposes of image-matching. This assumption may be violated should the re-projected view of the planar satellite image differ from the actual perspective view of the physical terrain. In order to study the differences of error in elevation projection, further work needs to be done with a satellite imagery textured digital elevation model of the terrain for in-depth studies.

6.2.4 Using Alternate Video Streams

This thesis was primarily assessing the effectiveness of using the day camera output of a UAV. Some UAVs however may also be equipped with IR cameras, which images the environment within a different spectral band. In terrains where the IMMAT algorithm may produce a poor estimation when using a day camera, the output could potentially be substituted with the view from the IR camera, which may reveal features that are otherwise imperceptible in daylight.

6.2.5 Studying the Effect of Actual Camera Field-of-View

Based on the data sets available, the Horizontal Field-of-View of 10.5° was used for King City recording and 35.26° for Camp Roberts recording. While the drop-rates seen in the King City flights were significantly higher than those for Camp Roberts, it is not possible to conclude whether it was the result of a smaller actual camera FOV or the effect of the terrain in King City that was challenging for the feature extraction algorithm to produce a match. Further studies using more data sets with varying actual camera fields-of-view are required to understand this aspect of the algorithm.

6.2.6 Using High-Fidelity Simulated Urban Environment Fly-By as Reference Images

There are systems that can generate high-fidelity simulations based on the inputs of a fly-through route. One example is that used by the Urban Redevelopment Authority of Singapore [19], which uses the system to visualize redevelopment plans ahead of time before approving any master plans (see **Figure 6.1**). While it is difficult to replicate the environment accurately for remote places, it should be possible to get a reasonably accurate model of a 3D urban environment. One pertinent research question is whether the image-matching algorithm still be able to provide reasonable estimates of position despite using a simulated scene as reference.

Figure 6.1. Simulation of an urban environment by Urban Redevelopment Authority of Singapore. Source: [19].

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A:
TASE 200 Output Data

The TASE200 sensor system bundles information with each frame captured, at $30Hz$. This appendix provides a description of the TASE200 sensor data format logged by the onboard sensor. **Table A.1** shows a comprehensive listing of all the meta-data that is captured by the TASE system.

Table A.1. TASE Meta-data available for analysis.

1	GPS Day	(byte 41)	25	Mount Roll	(bytes 260-263)
2	GPS Hour	(byte 42)	26	Mount Pitch	(bytes 264-267)
3	GPS Minute	(byte 43)	27	Mount Yaw	(bytes 268-271)
4	GPS Second	(bytes 44-47)	28	VN	(bytes 76-79)
5	Second since reset	(bytes 136-139)	29	VE	(bytes 80-83)
6	Second since midnight	(bytes 12-15)	30	VD	(bytes 84-87)
7	Gimbal Lat	(bytes 56-63)	31	Heading	(bytes 316-319)
8	Gimbal Lon	(bytes 65-71)	32	HFOV	(bytes 168-171)
9	Gimbal Alt	(bytes 72-75)	33	VFOV	(bytes 172-175)
10	Gimbal Pan	(bytes 24-271)	34	HFOVmax	(bytes 176-179)
11	Gimbal Tilt	(bytes 28-31)	35	HFOVmin	(bytes 180-183)
12	Gimbal Roll	(bytes 32-35)	36	Zoom	(bytes 186-187)
13	Image Lat	(bytes 192-199)	37	HFOVmaxC2	(bytes 212-215)
14	Image Lon	(bytes 200-207)	38	HFOVminC2	(bytes 216-219)
15	Image Alt	(bytes 208-211)	39	Transx	
16	Axis Pan Rate	(bytes 140-143)	40	Transy	
17	Axis Tilt Rate	(bytes 144-147)	41	GPS Satellites	(bytes 48-49)
18	Axis Roll Rate	(bytes 148-151)	42	GPS Status	(bytes 50-51)
19	Mount Pan Rate	(bytes 152-155)	43	GPS PDOP	(bytes 52-55)
20	Mount Tilt Rate	(bytes 156-159)	44	Magx	(bytes 310-311)
21	Mount Roll Rate	(bytes 160-163)	45	Magy	(bytes 312-313)
22	Roll	(bytes 88-91)	46	Magz	(bytes 314-315)
23	Pitch	(bytes 92-95)	47	Focus	(bytes 256-257)
24	Yaw	(bytes 96-99)			

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B:
Satellite Images Meta-data

This appendix summarizes the meta-data of the satellite images that were downloaded for King City.

Table B.1. Meta-data for the satellite tiles downloaded for King City.

Product Type	Panchromatic	Panchromatic	Panchromatic
Source	WV01	WV01	WV01
Source Unit	Strip	Strip	Strip
Ground Sample Distance	50 cm	50 cm	50 cm
NIIRS	4.7	4.8	4.9
Acquisition Date	2017-06-01 22:07 UTC	2017-07-18 21:58 UTC	2017-07-18 21:59 UTC
Cloud Cover	0.00%	0.00%	0.00%
Has Cloudless Geometry	Yes	Yes	Yes
Off Nadir Angle	28.6397°	24.9357°	16.9780°
Sun Elevation	59.3980°	61.9622°	61.8045°
Sun Azimuth	251.1663°	243.9128°	244.1939°
Data Layer	daily_take	daily_take	daily_take
Crs From Pixels	EPSG:4326	EPSG:4326	EPSG:4326
Precise Geometry	Yes	Yes	Yes
Per Pixel X	4.50E-06	4.50E-06	4.50E-06
Per Pixel Y	-4.50E-06	-4.50E-06	-4.50E-06
CE90 Accuracy	8.4	8.4	8.4
RMSE Accuracy	3.914259087	3.914259087	3.914259087
Spatial Accuracy	1:12,000	1:12,000	1:12,000

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C:
Schematic of MATLAB Program Flow

This appendix describes the flow of the MATLAB program. At a high level, the software is broken up into several major functional aspects stored in different files: `TracksDB.m`, `CreateSatelliteImageryAndTransforms.m`, `GenerateRawTrajectoryVideoClip.m`, `GenerateRawTrajectory.m`, `GenerateNominalTrajectory.m`, `GenerateReferenceFrames.m` and `ImageMatchingAlgorithm.m`.

TracksDB.m

After segmenting the raw tracks in the TASE videos into raw trajectories, this file records the starting and ending indices in TracksDB. The average above-ground-level altitude of each track, and its assigned track name are also stored in the database for easy reference in the rest of the program.

CreateSatelliteImageryAndTransforms.m

This script takes a folder of satellite image tiles and stitches them together into a large canvas. The script also computes the transform that maps each image pixel to UTM coordinates.

GenerateRawTrajectory.m

This function takes the starting and ending indices of the associated meta-data with the camera frames and pre-processes to the data to remove known biases.

GenerateRawTrajectoryVideoClip.m

This function takes the starting and ending indices from the TracksDB and assembles the separate frames into a video clip, that will be fed into the IMMAT algorithm.

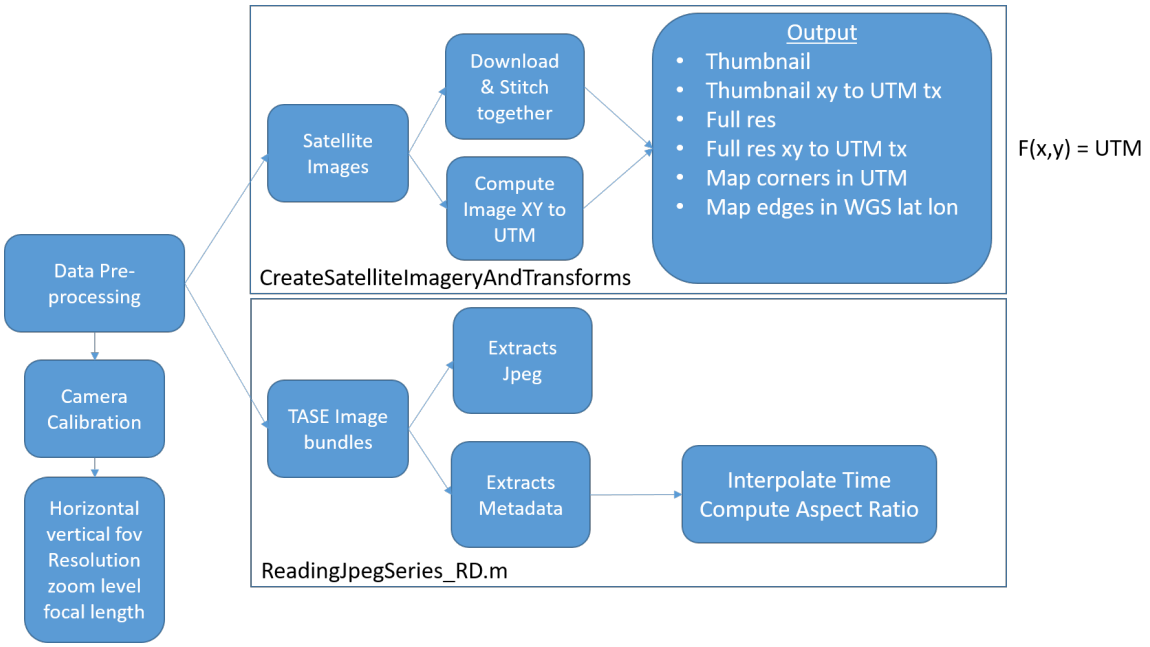

Figure C.1. Schematic for CreateSatelliteImageryAndTransforms.m

GenerateNominalTrajectory.m

This function generates a nominal trajectory based on the raw trajectory data by smoothening the data. The nominal trajectory contains position and pose information for a would be camera in flight to be used for generating reference frames.

GenerateReferenceFrames.m

This function takes the location and pose of a camera as described in the nominal trajectory and performs projective transformation of the top-down satellite view into the perspective view of the camera.

ImageMatchingAlgorithm.m

ImageMatchingAlgorithm is the core function that executes the IMMAT procedure. ImageMatchingAlgorithm.m first starts by extracting the features from both the reference image and the camera image, then doing a rough correspondence matching, and finally passing that information to `estimateGeometricTransform` which will cull outlier matches and then estimate a projective transformation between the reference frame and the camera image.

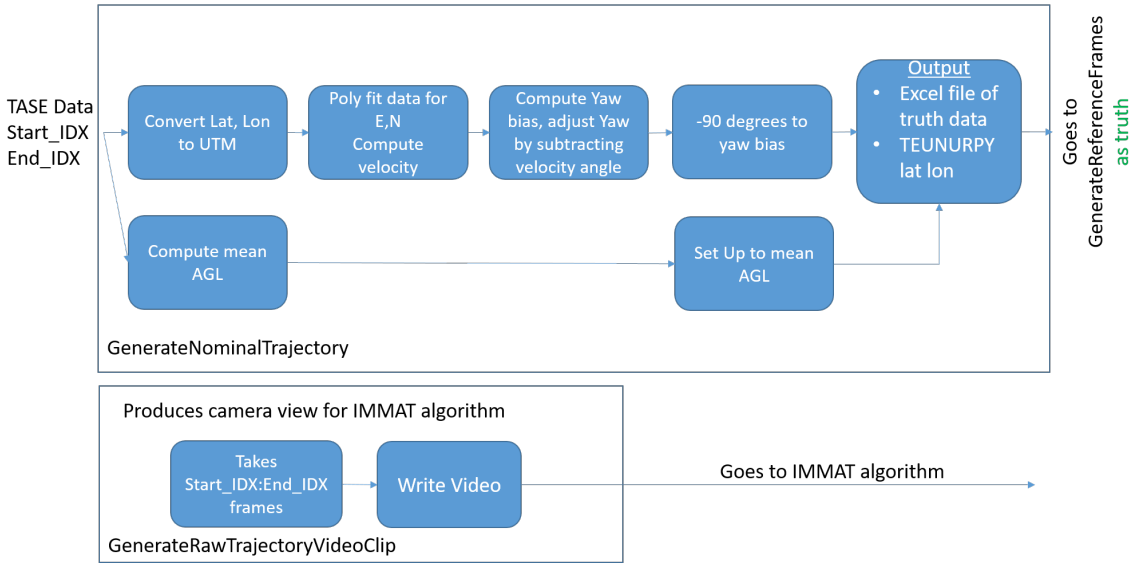

Figure C.2. Schematic for GenerateNominalTrajectory.m

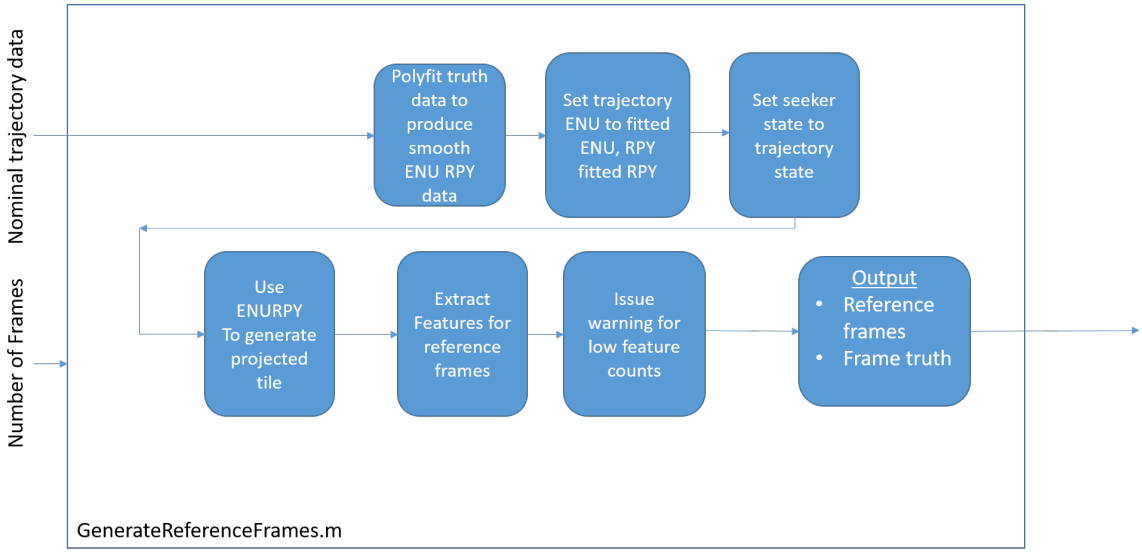

Figure C.3. Schematic for GenerateReferenceFrames.m

It then calls `estimateCameraPositionandOrientation` which projects the found inlier points of the camera image onto the ground plane and minimizes the displacement error between the observed points and re-projected points.

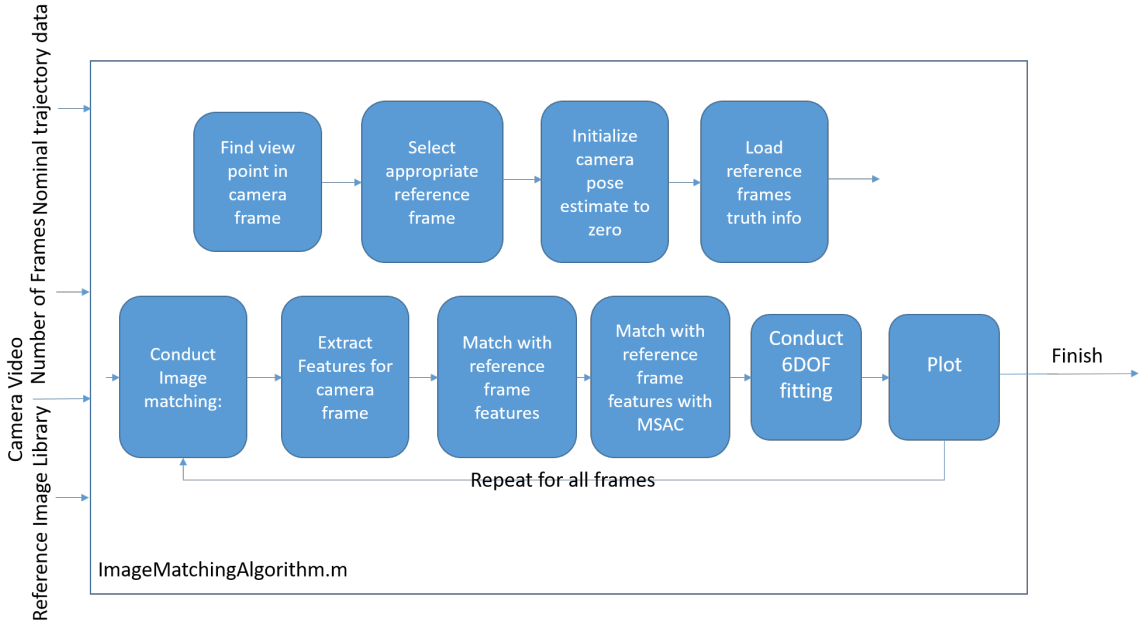

Figure C.4. Schematic for ImageMatchingAlgorithm.m

List of References

[1] D. Titterton and J. L. Weston, *Strapdown inertial navigation technology*. IET, 2004, vol. 17.

[2] W. Kong, G. Egan, and T. Cornall, "Feature-based navigation for UAVs," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*. IEEE, 2006, pp. 3539–3543.

[3] O. Yakimenko and R. Decker, "On the development of an image matching navigation algorithm for aerial vehicles," *Proceedings of the IEEE Aerospace Conference, Big Sky, MT*, 2016.

[4] U.S. Government. GPS Constellation Arrangement. [Online]. Available: http://www.gps.gov/systems/gps/space/. Accessed September 13, 2017.

[5] GPS Triangulation. [Online]. Available: http://gis.depaul.edu/shwang/teaching/geog258/GPS.htm. Accessed September 13, 2017.

[6] Department of Defense. (2011). The Unmanned Systems Integrated Roadmap FY2011-2036. [Online]. Available: https://my.nps.edu/documents/106607930/106914584/UxV+DoD+Integrated+Roadmap+2011.pdf/0f123fb1-ef1f-4842-9855-85a136b28a93. Accessed September 13, 2017.

[7] B. S. Blanchard and F. J. Benjamin, *Systems Engineering and Analysis*, 5th ed. Englewood Cliffs, NJ, Prentice Hall, 2010.

[8] K. W. Eure, C. C. Quach, S. L. Vazquez, E. F. Hogge, and B. L. Hill, "An application of UAV attitude estimation using a low-cost inertial navigation system," 2013.

[9] Omnidirectional Vision at University of Pennsylvania. (n.d.). University of Pennsylvania. [Online]. Available: http://www.cis.upenn.edu/~kostas/omni/. Sep 13, 2017.

[10] Omnidirectional Vision at University of Essex. (n.d.). University of Essex. [Online]. Available: http://cswww.essex.ac.uk/mv/images.html. Sep 13, 2017.

[11] I. F. Mondragón, P. Campoy, C. Martinez, and M. Olivares, "Omnidirectional vision applied to unmanned aerial vehicles (UAVs) attitude and heading estimation," *Robotics and Autonomous Systems*, vol. 58, no. 6, pp. 809–819, 2010.

[12] DigitalGlobe Satellite Imagery. (2017). [Online]. Available: https://www.digitalglobe.com/. Accessed 16 Apr 2017.

[13] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of ComputerVision*, vol. 60, no. 2, pp. 91–110, 2004.

[14] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *2011 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2011, pp. 2548–2555.

[15] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," *Computer Vision–ECCV 2006*, pp. 404–417, 2006.

[16] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[17] S. Choi, T. Kim, and W. Yu, "Performance evaluation of RANSAC family," *Journal of Computer Vision*, vol. 24, no. 3, pp. 271–300, 1997.

[18] R. Faragher, "Understanding the basis of the Kalman filter via a simple and intuitive derivation [lecture notes]," *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 128–132, 2012.

[19] "Channel News Asia, URA explores creating 3D digital models of city area using drones," May 2015. Available: http://www.channelnewsasia.com/news/singapore/ura-explores-creating-3d-digital-models-of-city-area-using-drone-8269966

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California